ELTE Eötvös Loránd University

Faculty of Science

Acyclic orientations of graphs

Thesis

Nóra Anna Borsik

Applied mathematics MSc

Supervisor:

Péter Madarasi

ELTE Institute of Mathematics

Department of Operations Research

Budapest

2024

Acknowledgements

I am deeply grateful to my supervisor, Péter Madarasi, for all the help and support he has provided me over the past few years. I am especially thankful for the endless and effective consultations, and for the incredible number of great ideas.

Contents

1 Introduction **1**

1.1 Motivation . . . 2

1.2 Our work . . . 3

1.3 Notations . . . 4

2 Related problems **4**

2.1 Acyclic orientations with degree constraints . . . 4

2.2 Minimizing the maximum (weighted) indegree . . . 5

2.2.1 Minimizing the maximum (weighted) indegree over orientations . . . 5

2.2.2 Minimizing the maximum (weighted) indegree over acyclic orientations . . . 7

2.3 Egalitarian orientations without acyclicity . . . 9

2.3.1 Path-reversal algorithm . . . 9

2.3.2 Generalization to M-convex sets . . . 11

2.4 The convex hull of the indegree vectors of (acyclic) orientations . . . 12

3 Lexicographically optimal acyclic orientations **15**

3.1 The dec-min and the inc-max acyclic orientation problems . . . 15

3.2 The inc-min and the dec-max (acyclic) orientation problems . . . 21

4 Minimizing $\sum_{v\in V} h(\varrho(v))$ over acyclic orientations **22**

4.1 Hardness results . . . 23

4.2 A special case: Minimizing $\sum_{v\in V} \varrho^2(v)$ over acyclic orientations . . . 26

4.2.1 Hardness for simple graphs . . . 27

4.2.2 Greedy approximation algorithm . . . 28

4.3 An exponential dynamic programming algorithm . . . 30

5 Maximizing $\sum_{v\in V} \varrho(v)\delta(v)$ over acyclic orientations **31**

5.1 NP-hardness . . . 31

5.2 Polynomial-time algorithm for graphs with maximum degree at most 3 . . . 32

5.3 Approximation ratio of orienting by a random permutation . . . 34

5.4 De-randomized approximation algorithm . . . 37

6 Open questions **39**

1 Introduction

The aim of this work is to investigate egalitarian orientation problems of graphs under the additional constraint that the orientation shall be *acyclic.* We put special emphasis on the following acyclic orientation problems, in which the indegree vector of an optimal orientation is required to be “smooth”, “equitable” or “egalitarian”.

Problem 1 (*dec-min* (decreasingly minimal) acyclic orientation) *Our goal is to find an acyclic orientation of a given graph $G = (V, E)$ in which the indegree vector sorted in non-increasing order is lexicographically minimal.*

Problem 2 (*inc-max* (increasingly maximal) acyclic orientation) *Our goal is to find an acyclic orientation of a given graph $G = (V, E)$ in which the indegree vector sorted in non-decreasing order is lexicographically maximal.*

Problem 3 ($\min \sum_{v\in V} h(\varrho(v))$ acyclic orientation) *Let us be given a graph $G = (V, E)$ and a discrete strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$ (i.e. $h(z+2)+h(z) > 2h(z+1)$ holds for every $z \in \mathbb{Z}_+$). Our goal is to find an acyclic orientation of G minimizing $\sum_{v\in V} h(\varrho(v))$, where $\varrho(v)$ denotes the indegree of $v \in V$.*

The non-acyclic counterparts of these problems are called egalitarian orientations in the literature, see [6, 12, 13], and they will be further discussed in Section 2.

The following orientation problem is significantly different from the previous egalitarian problems, since, instead of only focusing on the indegree vector, the optimal acyclic orientation minimizes the difference between the indegree vector and outdegree vector, in a certain sense. It can be seen as an acyclic version of the Eulerian and almost-Eulerian orientations, in which the difference of the indegree and outdegree of each vertex is at most 1.

Problem 4 ($\max \sum_{v\in V} \varrho(v)\delta(v)$ acyclic orientation) *Let us be given a graph $G = (V, E)$. Our goal is to find an acyclic orientation of G which maximizes $\sum_{v\in V} \varrho(v)\delta(v)$, where $\varrho(v)$ and $\delta(v)$ denote the indegree and the outdegree of v, respectively.*

It is important to note that each acyclic orientation problem above can be equivalently rephrased as a vertex ordering problem. Instead of acyclic orientations, we search an order of the vertices, and replace the indegrees ϱ with the left degrees $\overleftarrow{d}$, and the outdegrees δ with the right degrees $\overrightarrow{d}$ in the definition of the problems above. An order is optimal to the vertex ordering problem obtained this way if and only if the acyclic orientation obtained by orienting each edge from left to right is optimal to the original acyclic orientation problem. To see this, we need to observe that the indegree (and outdegree) vector of an acyclic orientation is the same as the left degree (and right degree) vector of its topological orderings. Throughout this work, we do not make a distinction between the acyclic orientation and the vertex ordering problems, although we mostly work with the vertex-ordering versions.

Now, we describe the structure of this work. In the rest of this section, we give some motivating applications to egalitarian orientation problems. After that, we give a brief summary of our work, and introduce the used notations. Section 2 summarizes related orientation problems, dealing with both acyclic and not necessarily acyclic orientations. First, we describe the known complexity results for finding an indegree-bounded acyclic orientation [18]. Then, we move on to the problem of minimizing the maximum (weighted) indegree in an orientation and in an acyclic orientation, which is the first step towards finding an egalitarian orientation of a graph. After that, we present the path-reversal algorithm from [6] for finding a not necessarily acyclic egalitarian orientation and briefly describe the more general framework introduced in [12, 13], which contains many different egalitarian orientation problems as special cases. Section 2.4 investigates the polyhedral characterization of the indegree vectors of (acyclic) orientations. After this review of related problems, we move on to the main topic of this work. From that point, we only consider the acyclic egalitarian orientation problems introduced at the beginning of Section 1. Section 3 considers lexicographically optimal acyclic orientation problems. In Section 3.1, we analyze the dec-min and inc-max acyclic orientation problems (Problem 1 and 2, respectively) even in the case when the indegrees are bounded by k, and in the unbounded case. In Section 3.2, we introduce the inc-min and dec-max problems, as the natural counterparts of these problem. In Section 4, we move on to minimizing $\sum_{v\in V} h_v(\varrho(v))$ of an acyclic orientation (Problem 3), and we put special emphasis on the special case of minimizing the square-sum of the indegrees in Section 4.2. Then, we analyze the problem of finding an acyclic orientation maximizing $\sum_{v\in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$ (Problem 4) in Section 5. In the end, we mention some open questions.

1.1 Motivation

Now, we show some applications of egalitarian orientation problems that we defined above. The first two problems are applications of egalitarian orientation problems without the acyclicity condition, and the routing protocol is an application of the egalitarian acyclic orientation problems (Problems 1- 3).

Road maintenance As the first motivating example, consider cities connected with roads to be maintained. We want to assign each road to one of the two cities it connects. The goal is to distribute the roads between the cities as evenly as possible. This problem is exactly an egalitarian orientation problem: Consider the graph $G = (V, E)$ in which the vertices correspond to the cities and the edges correspond to the roads. A road-distribution can be represented as an orientation. The road represented by the edge e is maintained by the city towards which the edge e is oriented. Therefore, the indegrees represent the number of roads maintained by the same city. Since we are searching for a fair distribution, the goal is to find an egalitarian (not necessarily acyclic) orientation of the graph.

Scheduling on a smart grid Another example comes from a scheduling problem described in [7]. They considered an offline scheduling problem in which customers send in unit time power requests and a set of time slots during which their requests can be served. For each time slot, the electricity cost is measured by a convex function of the load. The goal is to minimize the total electricity cost.

More formally, let us denote the set of unit time jobs by $J = \{j_1, \dots, j_n\}$. The time is divided into unit time time slots $T = \{t_1, \dots, t_m\}$, and each job j_i has a corresponding set of feasible time slots $T_i \subseteq T$. The number of requests assigned to a time slot t_k is called the *load* of t_k, and it is denoted by $\ell(t_k)$. There is given a convex cost function h, and our goal is to minimize $\sum_{t\in T} h(\ell(t))$.

This problem can be rephrased as an egalitarian orientation problem, on the following bipartite graph $G = (V, E)$. Let $V = J \cup T$ and add an arc $j_i t_k$ if $t_k \in T_i$, that is, if the job j_i can be processed at the time slot t_k. Consider an orientation of this bipartite graph, in which the outdegree of each job is equal to one (equivalently, the indegree of the vertex corresponding to the job j_i is equal to $(|T_i| - 1)$). In such an orientation, the indegree of the vertex corresponding to a time slot is equal to the load of that time slot (and the indegree of any other vertex, corresponding to a job, is fixed). Therefore, finding an optimal schedule is equivalent to finding an orientation of G with indegree prescription on the vertices corresponding to the jobs which minimizes $\sum_{v\in V} h(\varrho(v))$, which is an obvious generalization of Problem 3. For this problem a polynomial-time algorithm is given in [12], which is further discussed in Section 2.3.2.

Up-down routing The previous two examples were applications of egalitarian orientations without the acyclicity condition, but the next routing protocol is a practical application of the egalitarian acyclic orientation problems. In a routing problem, there is given an undirected graph called network. The vertices of the network represent switches with input buffers and the edges represent links. We want to send some packets between given start and end destinations. The goal is to give an efficient method to determine a route for every packet. In a wormhole routing, each packet is broken into small pieces, called flits. The first and last pieces are called heather flit and tail flit, respectively. The heather flit contains the routing information of the packet (for example the start and end destinations). The flits corresponding to one packet are routed continuously, this means that the heather tail reserves every reached buffer, and the buffer only works on the flits of that packet, until the tail flit arrives and frees the buffer. In a general step, every switch forwards the current flit from their input buffer to the next switch in the packet's route provided that the input buffer of that switch is not full. Such networks can suffer from deadlock, when there are several flits waiting along a cycle. An example for a deadlock is shown in Figure 1.

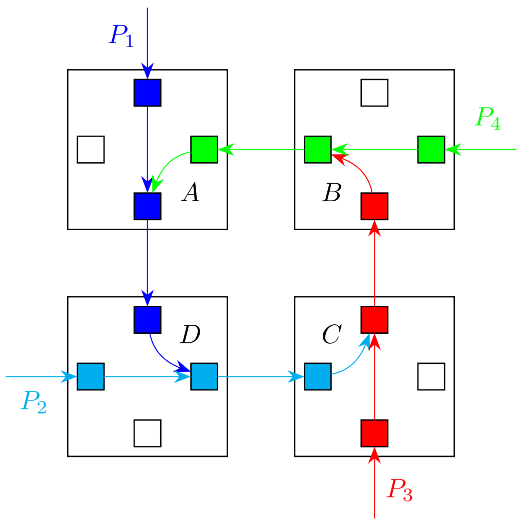


Figure 1: A deadlock with four packets.

The goal is to determine a route for each packet without causing a deadlock. It is known that deadlocks can be prevented by the following method called up-down routing [23, 24, 27]: Choose an acyclic orientation of the network, and forbid the packets to go through the sub-paths formed by any two edges directed into the same vertex.

Clearly, the number of forbidden sub-paths at a vertex only depends on the indegree of that vertex. Therefore, if we want to minimize the number of forbidden sub-paths, in some sense, then we have to choose the acyclic orientation in the up-down routing carefully. For example, if we choose a dec-min acyclic orientation then we can minimize the maximum number of forbidden sub-paths at any vertex. Similarly, considering a min $\sum_{v\in V} \varrho(v)^2$ acyclic orientation minimizes the total number of forbidden sub-paths. Moreover, if we only search for rooted-connected acyclic orientations then there exists a feasible route between every two switches, where an orientation is rooted-connected if there exists a root vertex from which any other vertex is reachable on a directed path in the orientation. Therefore, in the real life applications they always search for a rooted-connected acyclic orientation.

1.2 Our work

Section 3.1 considers the dec-min and inc-max acyclic orientation problems in the case when the indegrees are bounded by k, and in the unbounded case. We show that the optimal solutions for the dec-min and inc-max acyclic orientation problems differ from each other in the unbounded case, but in the case of indegree bound two, the optimal solutions for the two problems coincide. The dec-min acyclic orientation problem is known to be NP-hard in the case of indegree bound k for any odd $k \geq 5$, and in the unbounded case [6]. We extend this result for any $k \geq 3$, and prove the NP-hardness of the inc-max acyclic orientation problem in case of indegree bound k for any $k \geq 3$, and in the unbounded case. We introduce the natural counterparts of these problems, the inc-min and dec-max orientation problems, and prove that they are NP-hard both in case of acyclic and arbitrary orientations. Section 4 introduces the min $\sum_{v\in V} h(\overleftarrow{d}(v))$ ordering problem, which is equivalent to the min $\sum_{v\in V} h(\varrho(v))$ acyclic orientation problem, where $h : \mathbb{Z}_+ \to \mathbb{R}$ is a discrete strictly convex function. As one of our main results, we prove the NP-hardness of the problem for loop-free multigraphs for *any* discrete strictly convex function h. We show that the more general version of the problem, when a function h_v is given for each $v \in V$ instead of a single function h, is polynomial-time solvable if either each function h_v is linear or we allow non-acyclic orientations. Then, in Section 4.2, we consider the special case of this problem in which we want to find an order minimizing the square-sum of the left degrees. We extend the NP-hardness proof even for simple graphs, and analyze the approximation ratio of a simple greedy algorithm. After that, we give an exact dynamic programming algorithm for the more general problem of finding an order minimizing $\sum_{v\in V} h_v(\overleftarrow{d}(v))$ for some given (not necessarily convex) functions $h_v : \mathbb{Z}_+ \to \mathbb{R}$, for all $v \in V$. If the h_v functions can be evaluated in polynomial time, then the running time of the algorithm is $O(2^{|V|}\mathrm{poly}(|V|, |E|))$. In Section 5 we examine the max $\sum_{v\in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$ ordering problem (which is equivalent to the max $\sum_{v\in V} \varrho(v)\delta(v)$ acyclic orientation problem). First we use the results of [5, 16] to show that the problem is NP-hard, even for simple graphs with maximum degree at most four and it is polynomial-time solvable for simple graphs with maximum degree at most three. After that, we analyze the expected approximation ratio of a random order of the vertices for multigraphs, and give a deterministic algorithm achieving the same approximation guarantee.

1.3 Notations

Throughout this work, $G = (V, E)$ denotes an undirected simple graph or multigraph. Since we consider acyclic orientations a multigraph is assumed to be loop-free, unless stated otherwise. The *degree* of a vertex v is denoted by $d_G(v)$ or simply by $d(v)$, which counts the incident loops once. The maximum degree in G is denoted by $\Delta(G)$. The subgraph of $G = (V, E)$ induced by the subset $V' \subseteq V$ is denoted by $G[V']$, and the degree of $v \in V$ restricted to the vertices in V' is denoted by $d(v, V')$. Similarly, $D = (V, A)$ denotes a digraph. The *indegree* of the vertex v is denoted by $\varrho_D(v)$ or simply by $\varrho(v)$. The *outdegree* of a vertex v is denoted by $\delta_D(v)$ or simply by $\delta(v)$. We often denote an order of the vertices by $\sigma = \sigma_1, \dots, \sigma_n$, which means that the vertex $\sigma_i \in V$ is at the i^{th} position in the order. We denote the set of all permutations of the vertex set V by $\mathcal{S}_V$. The *left degree* of the vertex $v = \sigma_i$ in the order σ is equal to $d(v, \{\sigma_1, \dots, \sigma_i\})$, and it is denoted by $\overleftarrow{d}_\sigma(v)$ or simply by $\overleftarrow{d}(v)$. Similarly, the *right degree* of the vertex $v = \sigma_i$ in the order σ is equal to $d(v, \{\sigma_{i+1}, \dots, \sigma_n\})$, and it is denoted by $\overrightarrow{d}_\sigma(v)$ or simply by $\overrightarrow{d}(v)$. A vertex order is called *k-bounded* for some fixed integer k if $\overleftarrow{d}(v) \le k$, that is, the left degree of each vertex v is at most k in the order. The minimal number k for which G has a k-bounded order is denoted by $\overleftarrow{d}_{\min}(G)$, this is called the *degeneracy* of G. We denote the set of non-negative integers and non-negative real numbers by $\mathbb{Z}_+$ and $\mathbb{R}_+$. Let $[z]^+$ denote $\max\{z, 0\}$, and let $\mathbb{1}_A$ denote the *indicator* of the statement A, which is equal to one if A is true and equal to zero otherwise. Let χ_U denote the *characteristic vector* of the subset $U \subseteq V$. In case of a single-element subset $U = \{v\}$, we use the simplified notation χ_v.

2 Related problems

Orientation problems of graphs have been studied extensively in the literature, see [6, 11, 12, 13]. This section gives a summary about orientation problems related to egalitarian acyclic orientation, covering both acyclic and non-acyclic orientation problems.

2.1 Acyclic orientations with degree constraints

In [18], the authors considered the problem of finding an acyclic orientation with a given lower bound $f(v)$ for the indegree of v and a given lower bound $g(v)$ for the outdegree of v for each vertex v. More formally, they introduced the following problem: Let us be given a graph $G = (V, E)$ and two functions $f : V \to \mathbb{R}_+$ and $g : V \to \mathbb{R}_+$ with $f(v) + g(v) \le d(v)$ for each $v \in V$. The goal is to decide whether G has an *(f, g)-bounded acyclic orientation*, where an orientation is called (f, g)-bounded if $f(v) \le \varrho(v) \le d(v) - g(v)$ for each $v \in V$. As the definition suggests, the problem with lower bounds given for the indegree and outdegree of each vertex is essentially the same as finding an acyclic orientation with given lower and upper bounds for the indegrees of each vertex.

They proved that finding such an acyclic orientation is NP-hard in general. However, if there are only lower bounds given for the indegrees (i.e. $g(v) = 0$ for each $v \in V$), then it can be decided in polynomial time whether a feasible acyclic orientation exists, by a greedy algorithm which searches for the topological order of such an orientation and fixes the vertices from left to right. They do not consider the natural generalization of this problem in case of weighted graphs, but it is easy to see that the natural generalization of this greedy approach also works if there is a weight function $w : E \to \mathbb{R}_+$ and the lower bound $f : V \to \mathbb{R}_+$ is given for the weighted indegree of the vertices. Later, in Section 2.2.2 we are going to show a similar algorithm, for finding an acyclic orientation minimizing the maximum weighted indegree. The authors of [18] identified other polynomial-time solvable cases. They proved that we can find an (f, g)-bounded acyclic orientation if $f(v)g(v) = 0$ for each $v \in V$, that is, we can find an acyclic orientation with given indegree bounds, if for each vertex either only a lower or only an upper bound is given. Another solvable special case was the problem with $f(v) = d(v) - g(v)$, which is the problem with strict prescription for the indegree of each vertex. They also observed that the (f, g)-bounded acyclic orientation problem with $f(v) = g(v) = 1$ for each $v \in V \setminus \{s, t\}$ is equivalent to the so called *s-t numbering problem* [10]. In this problem, we are given a graph and two vertices s and t, and the goal is to find an order of the vertices in which the first vertex is s, the last vertex is t and all other vertices have at least one edge going to the left and at least one edge going to the right. The s-t numbering problem is known to be polynomial-time solvable [10]. However, the slightly more general case of the (f, g)-bounded acyclic orientation problem with bounds $f(v) = g(v) = 2$ for each $v \in V \setminus \{s, t\}$ was proven NP-complete [18].

2.2 Minimizing the maximum (weighted) indegree

When someone thinks about egalitarian orientations, the first property that comes to mind is to minimize the maximum indegree of the vertices. This optimization problem was extensively studied in the past, see [3, 8, 9, 26].

2.2.1 Minimizing the maximum (weighted) indegree over orientations

In case of not necessarily acyclic orientations, the problem was proven polynomial-time solvable in three different articles [3, 9, 26]. The main idea of the algorithm is that it begins with an arbitrary orientation, and reverses directed paths which reduce the indegree of a vertex with maximum indegree. Later in Section 2.3.1, we present essentially the same algorithm and the proof of a stronger theorem from [6] which also implies that the orientation given by the algorithm minimizes the maximum indegree.

Consider the natural generalization of this problem for weighted digraphs, when we are given a graph $G = (V, E)$ and a non-negative weight functions $w : E \to \mathbb{R}_+$ on the edge set. Our goal is find an orientation that minimizes the maximum weighted indegree of the vertices.

Unlike the unweighted case, this problem is NP-hard. We give a simple proof of the NP-hardness if parallel edges are allowed, and after that we present the proof given in [2] which also shows the NP-hardness in case of simple graphs.

Theorem 5 *Let us be given a multigraph $G = (V, E)$ and a non-negative weight functions $w : E \to \mathbb{R}_+$ on the edge set. It is NP-hard to find an orientation that minimizes the maximum weighted indegree of the vertices.*

PROOF: The proof is by reduction from the partition problem, which is known to be NP-complete [17]. In the partition problem, we are given m items with weights $a_1, \ldots, a_m$. Let $A = \sum_{i \in \{1,\ldots,m\}} a_i$ denote the sum of the weights. Our goal is to decide whether there exists a set of indices $I \subset \{1, \ldots, m\}$ such that $\sum_{i \in I} a_i = \frac{A}{2}$.

Let $a_1, \ldots, a_m$ be an instance of the partition problem. Construct the weighted graph $G = (V, E)$ az follows: Let G contain two vertices u and v, and m parallel uv arcs denoted by $e_1, \ldots, e_m$. Let the edge weights be $w(e_i) = a_i$ for $i \in \{1, \ldots, m\}$. Figure 2 shows the construction.

It is easy to see that the instance of the partition problem is solvable if and only if there exists an orientation of G in which the maximum weighted indegree is equal to $\frac{A}{2}$: First, if the set of indices I is a solution to the partition problem, then consider the orientation obtained by orienting the edge e_i from u to v if $i \in I$ and from v to u is $i \notin I$. Second, if there exists an orientation with maximum weighted indegree $\frac{A}{2}$, then let I denote the indices of those edges that are oriented from u to v. This is a solution to the partition problem. $\square$

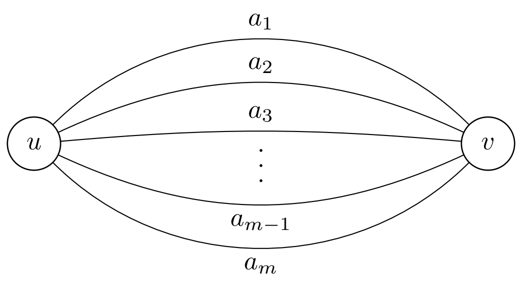


Figure 2: The graph constructed to the instance of the partition problem with weights $a_1, \dots, a_m$ in the proof of Theorem 5.

Note that in [3], the authors give a reduction to the partition problem which also proves the NP-hardness in case of simple bipartite planar graphs. Since the partition problem is weakly NP-complete, this only proves weak NP-hardness. However, in [2] the authors proved this problem to be strongly NP-hard, by the following reduction.

Theorem 6 (Asahiro, Jansson, Miyano, Ono, Zenmyo [2]) *Let us be given a graph $G = (V, E)$ and a non-negative weight functions $w : E \to \{1, k\}$, for any fixed $k \geq 2$. It is NP-hard to find an orientation that minimizes the maximum weighted indegree of the vertices.*

Proof: The proof is by reduction from the 3-SAT(2L) problem. In this problem, we are given conjunctive normal form (CNF) formula in which each clause contains at most 3 literals and each literal (not variable) occurs at most 2 times; and our goal is to decide whether it is satisfiable. The NP-hardness of this problem follows from [14].

Let us be given an instance of the 3-SAT(2L) problem with variables $v_1, \dots, v_n$ and clauses $c_1, \dots, c_m$, and construct the weighted graph $G = (V, E)$ as follows.

- For each variable x_i ($i \in \{1, \dots, n\}$), let G contain two vertices x_i and $\overline{x}_i$ corresponding to the literals of the variable x_i, and an edge $x_i\overline{x}_i$ with weight $w(x_i\overline{x}_i) = k$.
- For each clause c_j ($j \in \{1, \dots, m\}$), let G contain a vertex c_j, and for every literal x_i (or $\overline{x}_i$) which is in c_j, let G contain an edge c_jx_i (or $c_j\overline{x}_i$) with weight $w(c_jx_i) = 1$ (or $w(c_j\overline{x}_i) = 1$).
- Moreover, let G contain a cycle of length $\min\{3, k\}$, denoted by $C_{\min\{3,k\}}$, with edge weights equal to k, and connect the vertex c_j corresponding to the j^{th} clause with $(k + 1 - \#\{$ literals in $c_j\})$ different vertices of this cycle.

Figure 3 illustrates the construction.

We prove that the given instance of the 3-SAT(2L) problem is satisfiable if and only if G has an orientation in which the maximum weighted indegree of the vertices is at most k.

First, consider a satisfying truth assignment of the variables. For $i \in \{1, \dots, n\}$ if the variable x_i is true, then orient the edge $x_i\overline{x}_i$ from x_i to $\overline{x}_i$, otherwise, orient it to the other direction. For $j \in \{1, \dots, m\}$, choose a literal x_i (or $\overline{x}_i$) set to true contained in c_j and orient the edge c_jx_i (or $c_j\overline{x}_i$) towards x_i (or $\overline{x}_i$). Orient the remaining k edges incident to c_j towards c_j. Orient the cycle $C_{\min\{3,k\}}$ cyclically. In this orientation, each vertex x_i (or $\overline{x}_i$) corresponding to a literal set to true has indegree at most 2, because each literal occurs in at most 2 clauses; and each vertex x_i (or $\overline{x}_i$) corresponding to a literal set to false has weighted indegree equal to k. Each vertex c_j corresponding to a clause has weighted indegree equal to k, and each vertex of the cycle $C_{\min\{3,k\}}$ has also weighted indegree equal to k. So the maximum weighted indegree is k in this orientation.

Second, consider an orientation of G with maximum weighted indegree at most k. For each index $i \in \{1, \dots, n\}$, consider the edge $x_i\overline{x}_i$. If it is oriented towards x_i, then assign false to the variable x_i, otherwise, assign true to the variable x_i. Note that each vertex, corresponding to a false literal has already

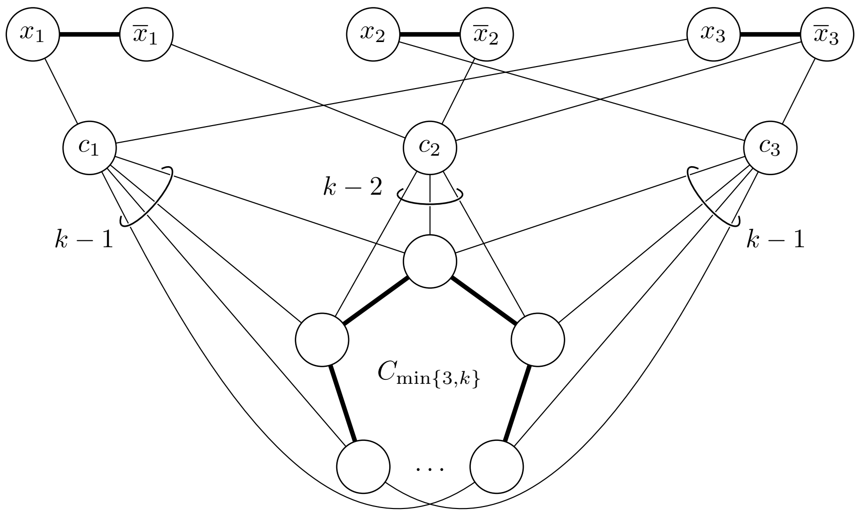


Figure 3: Example for the graph constructed in the proof of Theorem 6 for the CNF formula $(x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_2) \wedge (x_2 \vee \overline{x}_3)$. The thin edges have weight 1 and the thick edges have weight k.

indegree k, therefore, there are no other edges oriented towards these vertices. Observe that the edges of the cycle $C_{\min\{3,k\}}$ must be oriented cyclically; and all other edges incident to the vertices of $C_{\min\{3,k\}}$ must be oriented towards its other endpoint (which is a vertex c_j corresponding to some clause). Each vertex c_j corresponding to a clause has degree $k+1$, so they must have at least one edge oriented to its other endpoint. This can only be a vertex corresponding to a true variable, because we already observed that all other edges incident to c_j have to be oriented towards the vertex c_j. Therefore, each clause c_j contains at least one literal set to true, so the truth assignment satisfies the CNF formula. □

It also follows from the proof that we cannot decide whether the maximum weighted indegree is at most k or at least $k+1$. This implies the following theorem.

Theorem 7 (Asahiro, Jansson, Miyano, Ono, Zenmyo [2]) *Let us be given a graph $G = (V, E)$ and a non-negative weight functions $w : E \to \{1, k\}$, for a fixed $k \geq 2$. There exists no pseudo-polynomial time algorithm for finding an orientation minimizing the maximum weighted indegree of the vertices with approximation ratio less than $1 + \frac{1}{k}$, unless P = NP.*

Where an algorithm is pseudo-polynomial if it is polynomial in the numeric value of the input, but not necessarily in the length of the input.

2.2.2 Minimizing the maximum (weighted) indegree over acyclic orientations

The previous section discussed the problem of finding an orientation minimizing the maximum indegree of the vertices. It is natural to consider the analogous question in case of acyclic orientations. As we already mentioned in Section 1, finding an acyclic orientation with some restrictions for the indegrees is essentially the same as finding an order of the vertices with the same restrictions for the left degrees. So throughout this section, the goal is to find an order of the vertices such that the maximum (weighted) left degree of the vertices is minimized.

First, we consider the problem in case of unweighted graphs and describe a strongly related problem, for which an algorithm solving our acyclic orientation problem is known. In [20], the authors introduced the *degeneracy number* of graphs. A graph $G = (V, E)$ is called *k-degenerate* if $d_{G[V']}(v) \leq k$ holds for any induced subgraph $G[V']$ of G and any $v \in V'$. The degeneracy number of G denoted by $\overleftarrow{d}_{\min}(G)$, is the smallest integer k, for which G is k-degenerate. They also observed that a graph is k-degenerate if and only if there exists an order of the vertices in which the left degree of each vertex is at most k, these orders are called k-bounded orders. From this follows that there exists an acyclic orientation of G with maximum indegree at most k if and only if G is k-degenerate; and minimizing the maximum indegree is the same as computing the degeneracy number of G. It is known that the degeneracy number can be computed in linear-time, by constructing a $\overleftarrow{d}_{\min}(G)$-bounded order [21]. The algorithm constructs the order by repeatedly removing vertices with minimum degree and putting them at the rightmost free place in the order. Note that orienting each edge from left to right in such an order gives an acyclic orientation minimizing the maximum indegree.

Now, we consider the analogous problem for weighted graphs: Let us be given a graph $G = (V, E)$ and a non-negative weight function $w : E \to \mathbb{R}_+$ on the edge set. Our goal is to find an acyclic orientation minimizing the maximum weighted outdegree, which is essentially the same as finding an order of the vertices minimizing the maximum weighted left degree.

We give an algorithm for this problem, which is the natural generalization of the algorithm known for the unweighted case [21].

Algorithm 1 Weighted smallest last ordering

```
1: $V' := V$; $n := |V|$
2: Let $\sigma_1, \dots, \sigma_n$ denote the vertex order we are searching for
3: for $i = n, \dots, 1$ do
4:     Choose $\sigma_i \in \arg\min\{v \in V' : d^w_{G[V']}(v)\}$
5:     $V' := V' - \sigma_i$
6: end for
7: output $\sigma_1, \dots, \sigma_n$
```

Algorithm 1 fixes the vertices from right to left. In Line 4, a vertex with minimum weighted degree in the graph induced by the non-fixed vertices is selected, and placed at the last free position. Now, we show the correctness of Algorithm 1.

Theorem 8 *Let us be given a graph $G = (V, E)$ and non-negative weight function $w : E \to \mathbb{R}_+$. Algorithm 1 finds a vertex order minimizing the maximum weighted left degree.*

Proof: Let $\sigma = \sigma_1, \dots, \sigma_n$ denote the order given by Algorithm 1, and let $\sigma' = \sigma'_1, \dots, \sigma'_n$ be an arbitrary order of the vertices. We show that the maximum weighted left degree in σ is at most the maximum weighted left degree in σ'. Let σ_i be a vertex with maximum weighted left degree in σ, so $\sigma_i \in \arg\max_{v \in V} \overleftarrow{d}^w_\sigma(v)$. Consider the last vertex σ'_j in the order σ' from the vertices $\sigma_1, \dots, \sigma_i$. This means that $\{\sigma_1, \dots, \sigma_i\} \subseteq \{\sigma'_1, \dots, \sigma'_j\}$. Observe that $\overleftarrow{d}^w_\sigma(\sigma_i) \le \overleftarrow{d}^w_{\sigma'}(\sigma'_j)$, because

$$\overleftarrow{d}^w_\sigma(\sigma_i) = d^w(\sigma_i, \{\sigma_1, \dots, \sigma_i\}) \le d^w(\sigma'_j, \{\sigma_1, \dots, \sigma_i\}) \le d^w(\sigma'_j, \{\sigma'_1, \dots, \sigma'_j\}) = \overleftarrow{d}^w_{\sigma'}(\sigma'_j)$$

holds, where the first and last equations come from the definition of the weighted left degree. The first inequality is true, because $\sigma'_j \in \{\sigma_1, \dots, \sigma_i\}$ and the algorithm fixes σ_i from this set of vertices, which means that σ_i is a vertex with minimum weighted degree in the subgraph induced by $\{\sigma_1, \dots, \sigma_i\}$. The second inequality follows because the weight function w is non-negative and $\{\sigma_1, \dots, \sigma_i\} \subseteq \{\sigma'_1, \dots, \sigma'_j\}$. This means that the weighted left degree of σ'_j in the order σ' is at least the weighted left degree of σ_i in the order σ, which is the maximum weighted left degree in σ. This completes the proof. $\square$

This proves that we can find an acyclic orientation which minimizes the maximum weighted indegree of the vertices. In contrast, Theorem 6 states that the same problem is NP-hard in case of not necessarily acyclic orientations.

2.3 Egalitarian orientations without acyclicity

This section introduces the dec-min, inc-max, and max $\sum_{v \in V} h(\varrho(v))$ orientation problems, which are the same as Problems 1, 2 and 3, respectively, except that we do not require the orientation to be acyclic. Throughout this section, we refer to the non-acyclic versions of the problems, unless stated otherwise.

2.3.1 Path-reversal algorithm

In this section, we present and analyze the path-reversal algorithm for finding a dec-min orientation. The main idea of the algorithm for computing an orientation which minimizes the maximum indegree was essentially given in [3, 9, 26]. The authors of [6] realized that with path-reversals, we can get a dec-min orientation, which is a stronger property than just minimizing the maximum indegree.

Algorithm 2 PATH-REVERSAL ALGORITHM

```
1: Let $D$ be an arbitrary orientation of the input graph $G = (V, E)$.
2: while there is a directed path $P$ from $u$ to $v$ in $D$, with $\varrho(u) < \varrho(v) - 1$ do
3:     Choose $P$ such that $\varrho(u)$ is maximal.
4:     Update $D$ by reversing $P$.
5: end while
6: output $D$
```

First, we prove that the algorithm can be implemented in $O(|E|^2|V|)$ time, based on the argument given in [6]. Consider an iteration in which the algorithm reverses a path from u to v with $\varrho(v) = \ell$. Before the reversal of this path, let Q denote the set of vertices of indegree strictly greater than ℓ, and Q' denote the set of vertices from which there exists a path to a vertex in Q. Observe that no vertex of the path from u to v is in Q', otherwise, the algorithm would reverse a different path ending in a vertex with greater indegree. After reversing the path from u to v, Q is still the set of vertices of indegree strictly greater than ℓ and Q' is still the set of vertices from which there exists a path from u to v. This implies that after we reverse any path in which the last vertex has indegree equal to ℓ, the algorithm does not reverse a path to a vertex with higher indegree. Therefore, we can divide the algorithm into at most k stages, where k denotes the maximum indegree in the original orientation. In the $(k-\ell)^{\text{th}}$ stage, the algorithm reverses paths in which the last vertex has indegree ℓ. In a stage, there can be at most $|V|$ path reversals, and we can find a reversible path in linear time, for example, with depth-first search. This implies that the running time of the algorithm is $O(|E|^2|V|)$.

Now, we give the proof from [6] which shows that Algorithm 2 finds a dec-min orientation.

Theorem 9 (Borradaile [6]) *Let us be given a graph $G = (V, E)$. Algorithm 2 finds a dec-min orientation of G.*

PROOF: The proof is by showing that any dec-min orientation of G can be transformed into the output given by Algorithm 2 without changing the indegrees, using the following two operations:

- Reverse the orientation of a directed cycle, this operation is called *cycle reversal.*
- Reverse the orientation of a directed path from u to v, with $\varrho(u) = \varrho(v) - 1$, this operation is called *weak reversal.*

Observe that neither of these two operations modifies the decreasingly ordered vector of the indegrees. Therefore, any orientation which is reachable from a dec-min orientation using cycle reversals and weak reversals is also a dec-min orientation.

Let D_{OPT} denote a dec-min orientation of G and D_{PR} denote the orientation given by the path-reversal algorithm. The proof is by induction on $\alpha = \sum_{v \in V} |\varrho_{D_{\mathrm{OPT}}}(v) - \varrho_{D_{\mathrm{PR}}}(v)|$.

If $\alpha = 0$, then the indegree of each vertex v is the same in D_{OPT} and in D_{PR}. Consider the arcs of D_{OPT} that have the opposite orientation in D_{PR}. Reversing these arcs transforms D_{OPT} into D_{PR}. Each connected component induced by these arcs forms an Euler tour, which means that we can reverse these arcs using cycle reversals.

Otherwise, if $\alpha > 0$, then there are some vertices which have different indegrees in the two orientations. Let $V_{\neq}$ denote the set of these vertices, so $V_{\neq} = \{v \in V : \varrho_{D_{\mathrm{OPT}}}(v) \neq \varrho_{D_{\mathrm{PR}}}(v)\}$. Choose a vertex v from $V_{\neq}$ for which $\varrho_{D_{\mathrm{OPT}}}(v)$ is maximal, and if there is a choice among many such vertices, then choose v such that $\varrho_{D_{\mathrm{PR}}}(v)$ is maximal.

First, if $\varrho_{D_{\mathrm{OPT}}}(v) > \varrho_{D_{\mathrm{PR}}}(v)$, then consider the set of vertices from which v is reachable in the orientation D_{OPT}, denoted by U. So $U = \{u \in V : \text{there is a directed path from } u \text{ to } v \text{ in } D_{\mathrm{OPT}}\}$. Observe that

$$\sum_{u\in U} \varrho_{D_{\mathrm{OPT}}}(u) \leq \sum_{u\in U} \varrho_{D_{\mathrm{PR}}}(u) \tag{1}$$

holds, because $\sum_{u\in U} \varrho_{D_{\mathrm{OPT}}}(u)$ is equal to the number of edges in G induced by U, and $\sum_{u\in U} \varrho_{D_{\mathrm{PR}}}(u)$ also counts these edges and there can be other edges that are entering U in D_{PR}.

Notice that $v \in U$ and $\varrho_{D_{\mathrm{OPT}}}(v) > \varrho_{D_{\mathrm{PR}}}(v)$. This and the inequality (1) together imply that there exists a vertex $u \in U$ for which $\varrho_{D_{\mathrm{OPT}}}(u) < \varrho_{D_{\mathrm{PR}}}(u)$ holds. Observe that the choise of v implies that $\varrho_{D_{\mathrm{OPT}}}(u) < \varrho_{D_{\mathrm{OPT}}}(v)$ holds. Moreover, $\varrho_{D_{\mathrm{OPT}}}(u) \geq \varrho_{D_{\mathrm{OPT}}}(v) - 1$ holds, otherwise, reversing the path from u to v would give a better orientation to the dec-min orientation problem, which contradicts the optimality of the orientation D_{OPT}. So $\varrho_{D_{\mathrm{OPT}}}(u) = \varrho_{D_{\mathrm{OPT}}}(v) - 1$ and there exists a weakly reversible path from u to v in D_{OPT}. Reversing this path reduces α by 2.

Second, if $\varrho_{D_{\mathrm{OPT}}}(v) < \varrho_{D_{\mathrm{PR}}}(v)$, then consider the set of vertices from which v is reachable in the orientation D_{PR}, denoted by U. So let $U = \{u \in V : \text{there is a directed path from } u \text{ to } v \text{ in } D_{\mathrm{PR}}\}$. Notice that $\sum_{u\in U} \varrho_{D_{\mathrm{OPT}}}(u) \geq \sum_{u\in U} \varrho_{D_{\mathrm{PR}}}(u)$ holds. Since $\varrho_{D_{\mathrm{OPT}}}(v) < \varrho_{D_{\mathrm{PR}}}(v)$ holds, there exists a vertex $u \in U$ for which $\varrho_{D_{\mathrm{OPT}}}(u) > \varrho_{D_{\mathrm{PR}}}(u)$ holds. Moreover $\varrho_{D_{\mathrm{OPT}}}(u) \leq \varrho_{D_{\mathrm{OPT}}}(v)$ holds because of the choice of v. These together imply that $\varrho_{D_{\mathrm{PR}}}(u) < \varrho_{D_{\mathrm{OPT}}}(u) \leq \varrho_{D_{\mathrm{OPT}}}(v) < \varrho_{D_{\mathrm{PR}}}(v)$, which means that $\varrho_{D_{\mathrm{PR}}}(u) < \varrho_{D_{\mathrm{PR}}}(v) - 1$ and there is a directed path from u to v in D_{PR}. This contradicts the fact that D_{PR} is an orientation given by the path-reversing algorithm. □

Claim 10 *The correctness of Algorithm 2 implies that a dec-min orientation is optimal for any orientation problem in which*

(1) the objective function only depends on the sorted sequence of the indegrees, and

(2) reversing a path from u to v, with $\varrho(u) < \varrho(v) - 1$ in any orientation results a not worse orientation. (Moreover, if it always results a strictly better orientation, then the optimal orientations are exactly the dec-min orientations.)

Proof: Consider an orientation problem fulfilling (1) and (2). Begin the path reversal algorithm from any optimal orientation. Observe that (2) implies that the dec-min orientation obtained by the path reversal algorithm is also optimal for the problem. (Moreover, if any path reversal strictly improves, then the optimal orientation itself was a dec-min orientation.) By (1), it follows that any dec-min orientation is optimal to the considered orientation problem. □

For example, any dec-min orientation minimizes the difference of the largest indegree and the smallest indegree, but the latter problem can have optimal orientations other than the dec-min orientations. However, the optimal orientations for the problem of minimizing the difference of the largest indegree and the smallest indegree, after that the difference of the second largest indegree and second smallest indegree, and so on, are exactly the dec-min orientations.

Furthermore, Claim 10 also implies the following corollary which was already made in [13], because the inc-max problem and the min $\sum_{v\in V} h(\varrho(v))$ problem fulfill (1) and the stricter version of (2).

Corollary 11 (Frank, Murota [13]) *The optimal orientations for the dec-min, for the inc-max, and for the* $\min \sum_{v\in V} h(\varrho(v))$ *orientation problems are the same.*

Later, in Section 3.1 we are going to give an example which shows that the optimal orientations to the acyclic versions of the problems no longer coincide.

2.3.2 Generalization to M-convex sets

In Section 2.3.1, we described the algorithm given in [6] for finding a dec-min orientation. In [12, 13], Frank and Murota considered a generalization of this problem, in which the goal is to find a dec-min element of an *M-convex set.* We need the following definitions before defining M-convex sets. Let S be a finite non-empty ground set. For a vector $x \in \mathbb{R}^S$ and a subset $Z \subseteq S$, we use the notation $\widetilde{x}(Z) = \sum_{z \in Z} x(z)$. A set-function $b : 2^S \to \mathbb{R} \cup \{+\infty\}$ is called *submodular*, if $b(X) + b(Y) \geq b(X \cap Y) + b(X \cup Y)$ holds for any $X, Y \subseteq S$. A submodular integer valued set-function b for which $b(\emptyset) = \emptyset$ and $b(S)$ is finite, defines a *base-polyhedron* $B = \{x \in \mathbb{R}^S : \widetilde{x}(S) = b(S), \widetilde{x}(Z) \leq b(Z) \text{ for every } Z \subset S\}$. A base-polyhedron is known to be an integral polyhedron. An M-convex set $\ddot{B}$ is the set of integer points of an integral base-polyhedron, that is, $\ddot{B} = B \cap \mathbb{Z}^S$.

They noticed that Corollary 11 also holds in this more general framework.

Theorem 12 (Frank, Murota [13]) *Let us be given an M-convex set* ($\ddot{B}$)*. The dec-min elements of* ($\ddot{B}$)*, inc-max elements of* ($\ddot{B}$) *and* $\min_{m \in \ddot{B}} \sum_{s \in S} h(m(s))$*, elements of* $\ddot{B}$*, for any* $h : \mathbb{Z} \to \mathbb{Z}$ *strictly convex function, are the same.*

So every statement given for the dec-min elements of an M-convex set also applies for the inc-max elements and $\min_{m \in \ddot{B}} \sum_{s \in S} h(m(s))$ elements.

In [12], Frank and Murota gave the so called basic algorithm for finding a dec-min element of an M-convex set $\ddot{B}$. This basic algorithm runs in poly($|S|, b(S)$) time, but it is not strongly polynomial in general. In the graph orientation applications, $S = V$ and $b(S) \leq |E|$, therefore the basic algorithm is polynomial in these cases. It is important to emphasize that they also gave a strongly-polynomial algorithm for finding a dec-min element of an M-convex set $\ddot{B}$, but we only focus on the simpler basic algorithm, since it is also strongly polynomial in the orientation related applications. The main idea of the basic algorithm is the same as in Algorithm 2. It starts from an arbitrary element $m \in \ddot{B}$ and if there exist coordinates s and t such that $m(t) \geq m(s) + 2$ and $m + \chi_s - \chi_t \in \ddot{B}$, then we replace m by $m + \chi_s - \chi_t$ and repeat the same procedure. They proved that if no such improvement exists, then the current element $m \in \ddot{B}$ is dec-min.

Frank and Murota also gave various applications of decreasing minimization on M-convex sets, see in [12]. In what follows we focus on the extensions of the graph orientation problems.

Dec-min orientations First, observe that the indegree vectors of orientations form an M-convex set. This follows from the well known Orientation lemma of Hakimi.

Lemma 13 (Hakimi [15]) *Let* $G = (V, E)$ *be an undirected graph and* $m : V \to \mathbb{Z}_+$ *a vector with* $\widetilde{m}(V) = |E|$*.* G *has an orientation with indegree vector* m *if and only if* $\widetilde{m}(X) \leq e_G(X)$ *for every subset* $X \subseteq V$*, where* $e_G(X)$ *denotes the number of edges with at least one vertex in* X*.*

It is known that $e_G : 2^V \to \mathbb{Z}_+$ is a submodular function. This immediately implies that the indegree vectors of orientations are the integer elements of the base-polyhedron $B = \{x \in \mathbb{R}^V : \widetilde{x}(V) = |E|, \widetilde{x}(Z) \leq e_G(Z) \text{ for each } Z \subseteq V\}$, hence they form an M-convex set. This implies that we can find a dec-min orientation using the previously described basic algorithm. Observe that the basic algorithm applied to the base-polyhedron $B(e_G)$, that is, for finding a dec-min orientation is essentially the same as Algorithm 2.

They also considered capacitated orientations. In this problem there is given a positive integer $\ell(uv)$ for each edge uv which denotes the number of parallel edges between u and v. Our goal is to get a dec-min orientation, by orienting some of the edges between u and v towards u and the others towards v. The algorithm given for the uncapacitated case is not strongly-polynomial if $\ell(e)$ can be arbitrary large, but they also gave a strongly-polynomial method for this problem. Moreover, they also gave a polynomial-time algorithm for computing a cheapest dec-min element of an M-convex set with respect to a given cost function $c : S \to \mathbb{R}_+$. In case of orientations, this means that we can find a dec-min orientation, minimizing $\sum_{v \in V} c(v)\varrho(v)$.

Indegree-constrained dec-min orientations They also proved that the intersection of an M-convex set and the set $\{x \in \mathbb{R}^S : f(s) \leq x(s) \leq g(s)\}$ is also an M-convex set. This implies that we can compute a dec-min element among the elements of an M-convex set fulfilling a given lower and upper bound on each coordinate in strongly-polynomial time. As an application, we can solve the following indegree-constrained orientation problem: Let us be given a graph $G = (V, E)$ and lower and upper bound functions $f : V \to \mathbb{Z}_+$ and $g : V \to \mathbb{Z}_+$. The goal is to find an orientation in which $f(v) \leq \varrho(v) \leq g(v)$ holds, and the indegree vector is decreasingly minimal (or increasingly maximal, or minimizes $\sum_{v \in V} h(\varrho(v))$ for some strictly convex function h) among these orientations.

With the basic algorithm, we can find an indegree-constrained egalitarian orientation for a graph in polynomial time by repeatedly reversing paths from a vertex u to a vertex v, such that $\varrho(v) > \varrho(u) + 1$ and $\varrho(u) < g(u)$ and $\varrho(v) < g(v)$. This algorithm can be applied to solve the scheduling problem from [7], which we mentioned in Section 1.1.

Strongly connected orientations They also proved in [12] that the indegree vectors of strongly connected orientations of 2-edge connected graphs form an M-convex set. This implies that, using the basic algorithm, we can find a dec-min strongly connected orientation by repeatedly reversing paths from a vertex u to a vertex v such that there are at least two edge-disjoint paths from u to v and $\varrho(v) > \varrho(u) + 1$, until no such path remains. The same algorithm was first suggested in [6] for finding dec-min strongly connected orientation, but they only proved that the resulting orientation minimizes the maximum indegree. Later the authors of [28] gave a proof for the correctness using only graph-theoretical concepts.

Moreover, Frank and Murota also considered (k, ℓ)*-edge-connected orientations* (with respect to a root node r_0), that is, $\varrho(X) \geq k$ whenever $\emptyset \neq X \subseteq V - r_0$ and $\varrho(X) \geq \ell$ whenever $r_0 \in X \subseteq V$. They proved that if a graph has (k, ℓ)-edge-connected orientations then the indegree vectors of these orientations form an M-convex set [12]. Therefore, we can also find a dec-min (k, ℓ)-edge-connected orientation in strongly-polynomial time.

2.4 The convex hull of the indegree vectors of (acyclic) orientations

As we previously mentioned in Section 2.3.2, the integral elements of the polyhedron $B(e_G) = \{x \in \mathbb{R}^V : \widetilde{x}(V) = |E|, \widetilde{x}(Z) \leq e_G(Z) \text{ for each } Z \subset V\}$ are exactly the indegree vectors of the orientations of G. Since it is known that all base-polyhedra are integral polyhedra, this also means that the corner solutions of $B(e_G)$ are indegree vectors of orientations of G. Furthermore, it is a bounded polyhedron, therefore it is the convex hull of the indegree vectors of the orientations of G. By personal communication with András Frank, we get the following characterization of the corner solutions of $B(e_G)$.

Theorem 14 *Let us be given a graph $G = (V, E)$. The corner solutions of the base-polyhedron $B(e_G) = \{x \in \mathbb{R}^V : \widetilde{x}(V) = |E|, \widetilde{x}(Z) \leq e_G(Z)$ for each $Z \subset V\}$ are exactly the indegree vectors of the acyclic orientations of G.*

Proof: $B(e_G)$ is known to be the convex hull of the indegree vectors of orientations of G. Therefore, it is enough to prove that

(1) the indegree vector of every non-acyclic orientation can be written as a convex combination of other indegree vectors, and

(2) the indegree vectors of acyclic orientations cannot be written as a convex combination of other indegree vectors.

First, consider a non-acyclic orientation D of G with indegree vector $x = (\varrho(v_1), \dots, \varrho(v_n))$, where $v_1, \dots, v_n$ denote the vertices of G. We prove (1) by constructing orientations such that x can be written as a convex combination of the obtained indegree vectors. Notice that D contains a directed cycle C, and assume without loss of generality that the cycle C consists of the arcs $v_1v_2, \dots, v_{k-1}v_k, v_kv_1$ in this order. Construct k orientations $D^1, \dots, D^k$ as follows: D^j is the same as D, except that the

j^{th} arc of the cycle C is reversed for each $j \in \{1, \dots, n\}$. So in D_1 the arc v_1v_2 is reversed, in D^2 the arc v_2v_3, and so on, in D^k the arc v_kv_1 is reversed. Note that the indegree vector of D^j is $x^j = (\varrho(v_1), \dots, \varrho(v_{j-1}), \varrho(v_j)+1, \varrho(v_{j+1})-1, \varrho(v_{j+2}), \dots, \varrho(v_n))$ for each $j \in \{1, \dots, k-1\}$, and the indegree vector of D^k is $x^k = (\varrho(v_1)-1, \varrho(v_2), \dots, \varrho(v_{k-1}), \varrho(v_k)+1, \varrho(v_{k+1}), \dots, \varrho(v_n))$. Observe that $x = \frac{1}{k}\sum_{j=1}^{k} x^j$ holds, so the indegree vector of the non-acyclic orientation D can be written as a convex combination of the indegree vectors $x^1, \dots, x^k$ of $D^1, \dots, D^k$.

Second, we prove (2) by induction on $|V|$. Suppose that for any graph G' on $(|V|-1)$ vertices, the indegree vectors of acyclic orientations cannot be written as a convex combination of other indegree vectors. Consider an acyclic orientation D of the graph $G = (V, E)$ with indegree vector $x = (\varrho(v_1), \dots, \varrho(v_n))$. Let x be a convex combination of indegree vectors $x^1, \dots, x^k$ of orientations $D^1, \dots, D^k$, respectively. So $x = \sum_{j=1}^{k} \lambda_j x^j$ for some $\lambda_1, \dots, \lambda_k > 0$, $\sum_{j=1}^{k} \lambda_j = 1$. We show that $x^j = x$ for $j \in \{1, \dots, k\}$.

Since D is an acyclic orientation, it has a vertex of indegree 0. We can assume without loss of generality that this vertex is v_n. Consider the graph $G' = G - v_n$ and the orientations $D, D^1, \dots, D^k$ restricted to G'. Let us denote these orientations by $D', D^{1'}, \dots, D^{k'}$ and the corresponding indegree vectors by $x', x^{1'}, \dots, x^{k'}$, respectively. Observe that D' is an acyclic orientation of G' and $x' = \sum_{j=1}^{k} \lambda_j x^{j'}$ holds, because the vectors $x', x^{1'}, \dots, x^{k'}$ are the same as the first $(n-1)$ coordinates of the vectors $x, x^1, \dots, x^k$ minus the vector $(d(v_1, v_n), \dots, d(v_{n-1}v_n))$, respectively. Therefore, $x^{j'} = x'$ holds for each $j \in \{1, \dots, k\}$, because of the inductive hypothesis. Furthermore, this implies that $x^j = x$ for each $j \in \{1, \dots, k\}$. $\square$

This characterization gives further motivation to consider egalitarian acyclic orientations, since finding an optimal acyclic orientation to Problems 1- 3 is equivalent to finding a corner solution of the polyhedron $B(e_G)$ which is decreasingly minimal, increasingly maximal or minimizes the objective function $\sum_{v \in V} h(\varrho(v))$. Furthermore, the previous theorem also implies that we can find a minimizer among the indegree vectors of acyclic orientations for any linear objective function. Later in Theorem 32, we give a simpler, more effective method for finding an acyclic orientation whose indegree vector minimizes a linear objective function.

Moreover, we give another polyhedral characterization for the convex hull of indegree vectors for which the method given in [22] can be used to find a dec-min orientation, even in the case of indegree-constrained orientations.

Consider the incidence matrix of the following graph $G' = (V', E')$: Let $V' = V \cup E$ and let E' contain an edge between $e \in V' \cap E$ and $v \in V' \cap V$ if in the graph $G = (V, E)$, $v \in V$ is an endpoint of $e \in E$. Consider the incidence matrix $A_{G'}$ of the bipartite graph G', and extend the rows corresponding to vertices in $V' \cap V$ with $-I_{|V| \times |V|}$, which denotes the identity matrix of dimension $|V| \times |V|$, and each row corresponding to vertex in $V' \cap E$ with 0^V. Denote the resulting matrix by A. Clearly, A is a totally unimodular matrix. Let $y \in \mathbb{R}_+^{E'}$ denote the variables corresponding to the columns of the matrix $A_{G'}$ and $x \in \mathbb{R}_+^V$ denote the variables corresponding to the additional columns containing -1 and 0 coordinates. Let $b = (0^V, 1^E)$. We claim the following.

Theorem 15 *Consider the polyhedron $P = \{(y, x) \in \mathbb{R}_+^{E'} \times \mathbb{R}_+^V : A(y, x) = b, (y, x) \geq 0\}$. The projection of P on the variables x, that is, $P|_x = \{x \in \mathbb{R}_+^V : (y, x) \in P$ for some $y \in \mathbb{R}^{E'}\}$, is the convex hull of the indegree vectors of orientations of G.*

Proof: Observe that P and $P|_x$ are bounded polyhedra. Therefore, they are the convex hull of their corner solutions. We show that every corner solution (y, x) of P corresponds to an orientation of G with indegree vector $x \in \mathbb{R}_+^V$, and for every corner solution x of $P|_x$, there exists $y \in \mathbb{R}_+^{E'}$ such that (y, x) is a corner solution of P.

Observe that A is a totally unimodular matrix and b is an integral vector, this implies that the corner solutions of P are integral vectors. Consider an integral vector $(y, x) \in P$. For any edge $e = uv \in G$, the equality $y_{eu} + y_{ev} = 1$ means that one of y_{eu}, y_{ev} is equal to one and the other one is equal to zero. Orient e towards u if $y_{eu} = 1$, and towards v otherwise. Observe that the indegree vector of this orientation is x, because of the equality $x_v = \sum_{e=uv} y_{ev}$. So every corner solution (y, x) of P corresponds to an orientation of G with indegree vector x.

Consider a corner solution $x \in P_x$. Suppose for contradiction that there exists no $y \in \mathbb{R}_+^{E'}$ for which (y, x) is a corner solution of P. Therefore, (y, x) can be written as a convex combination of some corner solutions $(y_1, x_1), \dots, (y_k, x_k) \in P \setminus \{(y, x)\}$. But then x can be written as the same convex combination of $x_1, \dots, x_k \in P|_x$. Since x was a corner solution in $P|_x$, this implies that $x_1 = x, \dots, x_k = x$. Therefore, $(y_1, x_1) = (y_1, x)$ is a corner solution in P, which contradicts the indirect assumption.

This implies that $P|_x$ is the convex hull of the indegree vectors of the orientations of G, which was to be shown. □

Notice that both $P|_x$ and the base-polyhedron $B(e_G)$ define the convex hull of the indegree vectors of orientations of G. Therefore, Theorem 14 implies the following.

Corollary 16 *The corner solutions of $P|_x$ are exactly the indegree vectors of the acyclic orientations.*

Furthermore, it follows from the conditions that any integral vector in the convex hull is an indegree vector of an orientation. This implies that the algorithm in [22] can be used to find a dec-min orientation. This algorithm naturally solves a more general orientation problem, in which we are given a graph $G = (V, E)$ along with a discrete convex function $h_v : \mathbb{Z}_+ \to \mathbb{R}$ for each $v \in V$. The goal is to find an orientation of G that minimizes $\sum_{v \in V} h_v(\varrho(v))$. Corollary 11 implies that we can solve the dec-min problem when we set h_v to the same strictly convex function h for every $v \in V$. But it is easy to see that if the h_v functions differ from each other, then the optimal solutions are not necessarily the dec-min orientations anymore.

In the rest of this section, we give another polynomial-time method for the generalized problem of finding an orientation of a graph G that minimizes $\sum_{v \in V} h_v(\varrho(v))$, where h_v is a given strictly convex function for each $v \in V$. We have to assume that the function h_v can be evaluated in polynomial time. We reduce this problem to the minimum cost flow problem, for which there exists a strongly-polynomial algorithm [25].

Theorem 17 *Let us be given a multigraph $G = (V, E)$ and a discrete strictly convex function $h_v : \mathbb{Z}_+ \to \mathbb{R}$ for each $v \in V$. We can compute an orientation minimizing $\sum_{v \in V} h_v(\varrho(v))$ in polynomial time provided that the function h_v can be evaluated in polynomial time for each $v \in V$.*

Proof: We reduce this problem to the minimum cost integral flow problem [25]. Let $G = (V, E)$ denote the multigraph for which we want to find an optimal orientation. Let $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_m\}$. If $h_v(0) \neq 0$, then we modify the function to $h'_v(z) = h_v(z) - h_v(0)$ for $z \in \mathbb{Z}_+$. Observe that $\sum_{v \in V} h_v(\varrho(v)) = \sum_{v \in V} h'_v(\varrho(v)) + \sum_{v \in V} h_v(0)$ for any orientation, so the optimal orientations are the same for the functions h'_v and h_v. Hence, we assume without loss of generality that $h_v(0) = 0$ for each vertex $v \in V$.

Construct the network $D = (V', A)$ as follows. Let V' contain a source vertex s, a sink vertex t, a vertex v'_i for each $i \in \{1, \dots, n\}$, and also a vertex e'_j for each $j \in \{1, \dots, m\}$. Let D contain $d_G(v_i)$ parallel arcs $a^1_\ell, \dots, a^{d(v_i)}_\ell$ from s to v'_i for $i \in \{1, \dots, n\}$. Add an arc from v'_i to e'_j if e_j is incident to v_i in G for $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$. Finally, add an arc from e'_j to t for every $j \in \{1, \dots, m\}$. Figure 4 illustrates the construction.

Let the capacity of each arc be 1. Set the cost $c(a^i_\ell)$ of the arc a^i_ℓ to $(h_{v_i}(\ell) - h_{v_i}(\ell - 1))$ for each $i \in \{1, \dots, n\}$ and each $\ell \in \{1, \dots, d(v_i)\}$. Let the cost $c(a)$ be 0 for the rest of the arc. Observe that the sum of the costs of the first z parallel sv'_i arcs $a^1_\ell, \dots, a^z_\ell$, for some $v_i \in V$ is

$$\sum_{\ell=1}^{z} a^i_\ell = \sum_{\ell=1}^{z} (h_{v_i}(\ell) - h_{v_i}(\ell - 1)) = h_{v_i}(z) - h_{v_i}(0) = h_{v_i}(z).$$

We prove that the minimum cost of an integral s-t flow of amount $|E|$ is equal to the $\sum_{v_i \in V} h_{v_i}(\varrho(v_i))$ of an optimal orientation of G.

On the one hand, consider an optimal orientation, and construct an s-t flow $f : A \to \mathbb{Z}_+$ of amount $|E|$ as follows. For all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$, let $f(a^i_\ell) = 1$ if $\ell \leq \varrho(v_i)$. Let $f(v'_i e'_j) = 1$

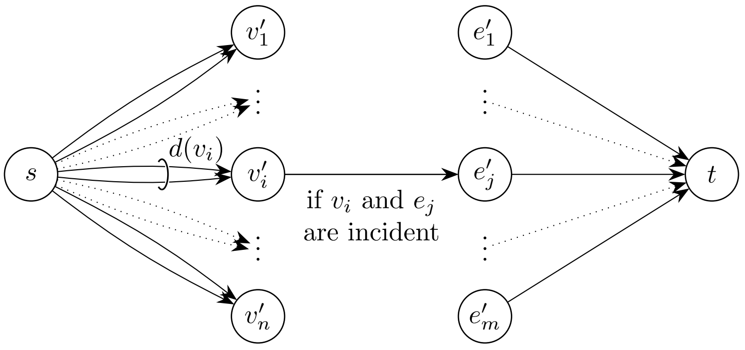


Figure 4: The digraph $D = (V, A)$ constructed in the reduction to the minimum cost flow problem. The vertices $v'_1, \dots, v'_n$ are corresponding to the vertices of G and the vertices $e'_1, \dots, e'_m$ are corresponding to the edges of G.

if $e_j \in E$ is oriented toward $v_i \in V$ in G. Set $f(e'_j t) = 1$. For the remaining arcs, let $f(a) = 0$. The resulting flow f is clearly feasible, and it uses the first $\varrho(v_i)$ parallel sv'_i arcs, therefore the cost of the flow is equal to $\sum_{v_i \in V} h_{v_i}(\varrho(v_i))$.

On the other hand, compute a minimum cost integral flow $f : A \to \mathbb{Z}_+$ of amount $|E|$. Note that f uses the arc $e'_j t$ for all $j \in \{1, \dots, m\}$, therefore, each $e'_j \in V'$ has exactly one incoming arc with $f(v'_i e'_j) = 1$. Orient each $e_j \in E$ towards the vertex $v_i \in V$ for which $f(v'_i e'_j) = 1$. Observe that the indegree $\varrho(v_i)$ of $v_i \in V$ is equal to the number of parallel sv'_i arcs used in the flow, since the function h_{v_i} is strictly convex, $c(a^i_\ell) = h_{v_i}(\ell) - h_{v_i}(\ell - 1) < h_{v_i}(\ell') - h_{v_i}(\ell' - 1) = c(a^i_{\ell'})$ holds whenever $\ell < \ell'$. Therefore, the flow uses the first $\varrho(v_i)$ parallel sv'_i arcs for each $i \in \{1, \dots, n\}$. This implies that the cost of the flow is $\sum_{v_i \in V} h_{v_i}(\varrho(v_i))$, which is the objective value of the constructed orientation of G.

This means that we can find an orientation minimizing $\sum_{v_i \in V} h_{v_i}(\varrho(v_i))$ in polynomial time by computing a minimum cost integral s-t flow in the network constructed above. As the latter problem is solvable in strongly-polynomial time, the theorem follows. $\square$

3 Lexicographically optimal acyclic orientations

Now after the overview of related problems, we begin to investigate the egalitarian acyclic orientation problems defined in Section 1. This section investigates the complexities of Problem 1 and 2, and related problems of finding a lexicographically optimal acyclic orientation. First, we describe the dec-min and inc-max ordering problems, which are essentially the same as Problem 1 and 2, respectively, and we prove them NP-hard. After that, we introduce the inc-min and dec-max acyclic orientation problems as natural counterparts of the dec-min and inc-max acyclic orientation problems.

3.1 The dec-min and the inc-max acyclic orientation problems

Let us be given a graph $G = (V, E)$. A vertex order is optimal for the dec-min (decreasingly minimal) problem if the sequence of the left degrees sorted in non-increasing order is lexicographically minimal. Similarly, a vertex order is optimal for the inc-max (increasingly maximal) problem if the sequence of the left degrees sorted in non-decreasing order is lexicographically maximal. An order is called k-bounded if the left degree of each vertex is at most k. The dec-min or inc-max ordering problems are equivalent to finding a dec-min or inc-max acyclic orientation of G, respectively. Corollary 11 states that without acyclicity the dec-min and inc-max orientation problems are equivalent. So the question arises whether these problems are also equivalent in the acyclic case. With a program, we found that the simple graph on 9 vertices shown in Figure 5 is the smallest example for which the dec-min and inc-max ordering problems are different.

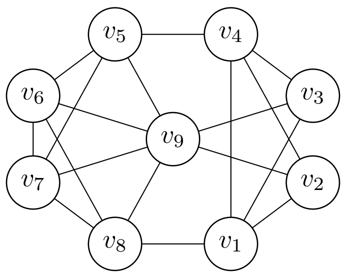


Figure 5: The smallest simple graph for which the dec-min and inc-max problems are not the same.

Claim 18 *For the simple graph shown in Figure 5, the sets of the optimal orders for the dec-min and the inc-max ordering problems are disjoint.*

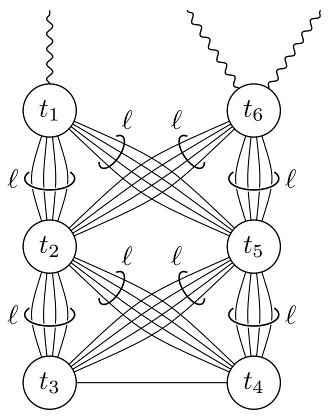
PROOF: We claim that the non-increasingly ordered left degree sequence of an optimal order for the dec-min problem is $3,3,3,3,2,2,1,1,0$ and the non-decreasingly ordered left degree sequence of an inc-max order is not worse than $0,1,2,2,2,2,2,3,4$, hence the optimal orders are not the same. First, notice that there exist 3-bounded orders for this graph, therefore, a dec-min order is also a 3-bounded order. Furthermore, in any 3-bounded order, at least two vertices of $\{v_5, v_6, v_7, v_8, v_9\}$ have left degree 3, because in any 3-bounded order for their induced subgraph the last two vertices have left degree at least 3. Moreover in any 3-bounded order, the last two vertices can only be from $\{v_1, v_2, v_3, v_4\}$, because only the vertices v_2 and v_3 have degree at most 3 and after deleting one of these vertices, the only new vertex of degree at most 3 is v_1 or v_4. These imply that in any 3-bounded order, the last two vertices among $\{v_1, v_2, v_3, v_4\}$ have left degree 3 and at least two other vertices among $\{v_5, v_6, \ldots, v_9\}$ have left degree at least 3. Together with the fact that the first vertex has left degree 0, this implies that the non-increasingly ordered degree sequence of a dec-min order is lexicographically at least as big as $3,3,3,3,2,2,1,1,0$. So the order $v_9, v_8, v_7, v_6, v_5, v_4, v_3, v_1, v_2$ (which has non-increasingly ordered degree sequence $3,3,3,3,2,2,1,1,0$) is optimal for the dec-min problem. Note that the order $v_1, v_4, v_2, v_3, v_9, v_5, v_6, v_7, v_8$ has non-decreasingly ordered left degree sequence $0,1,2,2,2,2,2,3,4$, which is lexicographically greater than the non-decreasingly ordered left degree sequence of a dec-min order (that is $0,1,1,2,2,3,3,3,3$). So for the graph shown in Figure 5, the dec-min and inc-max optimal orders are not the same, moreover there is no common optimal solution for the dec-min and the inc-max ordering problems. □

Notice that a dec-min order is always a $\overleftarrow{d}_{\min}(G)$-bounded order, but the previous example shows that this is not necessarily true for the inc-max orders.

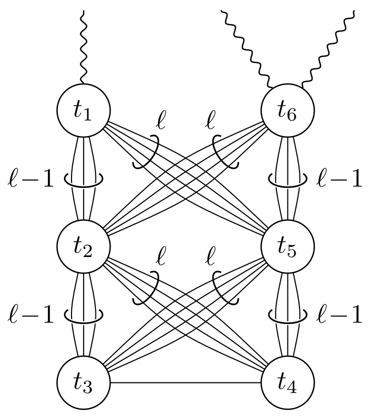
Now, we show that the dec-min k-bounded and inc-max k-bounded orders are the same for $k = 2$, but the complexity of these problems remains open. After that, we examine the complexities of finding a dec-min or inc-max k-bounded order, and prove them NP-hard for any $k \geq 3$ and if there is no bound on the left degrees. The NP-hardness of finding the dec-min k-bounded order was already proven for all odd $k \geq 5$ and in the case when there is no bound on the left degrees in [6]. We prove that the problem is hard for all $k \geq 3$. Then, we show that finding an inc-max k-bounded order is also NP-hard for any $k \geq 3$, and even if there is no bound on the left degrees.

Theorem 19 *Let us be given a graph $G = (V, E)$. For $k = 2$, the dec-min and inc-max k-bounded orders are the same.*

PROOF: Notice that in any 2-bounded order, the left degree of every vertex is 0, 1 or 2, and the sum of the left degrees is equal to $|E|$. The dec-min case is equivalent to finding a 2-bounded order minimizing the number of vertices with left degree 2, and the inc-max problem is equivalent to finding a 2-bounded order minimizing the number of vertices with left degree 0. Formally, let σ denote an order of the vertices and let n_i^σ denote the number of vertices with left degree equal to i. Then for any 2-bounded order



(a) The case $k = 2\ell + 1$.



(b) The case $k = 2\ell$.

Figure 6: The gadgets corresponding to $v \in V$ for the proof of Theorem 20.

$n_1^\sigma = |V| - n_0^\sigma - n_2^\sigma$ and $|E| = n_1^\sigma + 2n_2^\sigma = |V| - n_0^\sigma + n_2^\sigma$ holds. Consider two orders σ and σ', then $|E| = |V| - n_0^\sigma + n_2^\sigma = |V| - n_0^{\sigma'} + n_2^{\sigma'}$ holds, therefore, $n_2^\sigma - n_2^{\sigma'} = n_0^\sigma - n_0^{\sigma'}$. This implies that an order minimizes the number of vertices with left degree 2 if and only if it minimizes the number of vertices with left degree 0, therefore, the inc-max and dec-min k-bounded orders are the same. $\square$

Now we move on to the problems with larger bounds k. The dec-min k-bounded ordering problem was already proven NP-hard for every odd $k \geq 5$ parameters in [6]. The following theorem extends this result for any $k \geq 3$.

Theorem 20 *For any integer $k \geq 3$, it is NP-hard to find a k-bounded order that minimizes the number of vertices with left degree exactly k, even for simple graphs.*

Proof: First, we give the proof for multigraphs, then, we show how to eliminate parallel edges. The proof is by a reduction from the vertex cover problem on 3-regular simple graphs, which is NP-hard [1]. The construction is slightly different depending on the parity of k. If $k = 2\ell + 1$ for some $\ell \geq 1$, then let H denote the gadget shown in Figure 6a; otherwise, if $k = 2\ell$ for some $\ell \geq 2$, then let H be the gadget shown in Figure 6b. For a given 3-regular instance $G = (V, E)$ of the vertex cover problem, construct a graph $G' = (V', E')$ as follows. For each vertex v in G, add a disjoint copy H_v of H to G'. For every edge uv of G, add a new edge to G' between the vertices $t_1 \in H_u$ or $t_6 \in H_u$, and $t_1 \in H_v$ or $t_6 \in H_v$ in such a way that, in every gadget, exactly one and two new edges are incident to the vertices t_1 and t_6, respectively. The wavy edges at the top of Figures 6a and 6b illustrate these new edges, which correspond to the edges in G, and will be referred to as original (wavy) edges.

We show that the size of the smallest vertex cover in G equals the minimum number of vertices with left degree k in the k-bounded orders of the vertices of G'. The construction given above was different for odd and even k, but the rest of the proof is given for the two cases simultaneously.

Firstly, let $X \subseteq V$ denote a minimum-size vertex cover in G. We construct a k-bounded order for G' in which the number of vertices with left degree k is at most $|X|$. For each $v \in X$, remove t_3 from the gadget H_v in G', and then recursively remove every vertex of degree smaller than k. Whenever a vertex is removed, place it to the last free position in the order, from right to left. We prove that by the end of the procedure, every vertex is removed, and hence an order of V' is obtained. Take a vertex $v' \in V'$, and let $v \in V$ denote the vertex for which $v' \in H_v$. If $v \in X$, then the vertex t_3 in H_v is removed from G', which also causes v' to be removed by the construction of the gadgets. Otherwise, if $v \notin X$, then any edge $uv \in E$ incident to v must be covered by its other endpoint, that is, $u \in X$. But then the gadget H_u is removed, because its vertex t_3 is removed, which means that all three original (wavy) edges incident to H_u are removed. This means that all original (wavy) edges incident to H_v must be removed as well, hence the whole gadget H_v is removed by its construction, including the vertex v'. Therefore, all vertices in G' are removed, and hence we obtain an order of the vertices of G'. The left degree of every vertex is at most k in this order, because the degree of any vertex is at most k when it is removed. The only way a vertex of degree k can get removed is when t_3 is removed from the gadget H_v in G' for some $v \in X$, therefore, at most $|X|$ vertices have left degree k. This means that the order found by the procedure above is a k-bounded order, and there are at most $|X|$ vertices with left degree k.

Secondly, given a k-bounded order for G', we construct a vertex cover $X \subseteq V$ of G whose size is the number of vertices with left degree k. Let the set $X \subseteq V$ contain a vertex $v \in V$ if and only if there exists at least one vertex in H_v whose left degree is k in the k-bounded order. We prove that for any $uv \in E$, if $u \notin X$, then u' precedes v' in the k-bounded order, where $u'v' \in E'$ is the only original (wavy) edge connecting H_u and H_v. Note that u' is either $t_1 \in H_u$ or $t_6 \in H_u$, and v' is either $t_1 \in H_v$ or $t_6 \in H_v$. The last two vertices among $t_1, \dots, t_6 \in H_u$ in any k-bounded order must be t_1 and t_6, because $u \notin X$ means that there is no vertex in H_u with left degree k, and the vertices $t_2, \dots, t_5 \in H_u$ would have degree at least k if they appeared as any of the last two vertices by the construction of the gadgets. The only way $t_1 \in H_u$ and $t_6 \in H_u$ can have degree smaller than k is that the other endpoints of all incident original (wavy) edges are later in the order, hence u' must precede v'.

This immediately implies that X is a vertex cover, otherwise, there exists an edge $uv \in E$ such that $u, v \notin X$, but then the previous argument implies that u' precedes v' and also that v' precedes u', where u' and v' are the endpoints of the original (wavy) edge connecting H_u and H_v in G', which is not possible. Furthermore, the size of X is exactly the number of vertices with left degree k, which completes the proof of the theorem for multigraphs.

Now, we modify the construction so that the resulting graph becomes simple. For $\ell \geq 1$ and $k \in \{2\ell, 2\ell+1\}$, consider the corresponding gadget H shown in Figure 6. For any $p \in \{\ell-1, \ell\}$ and any two distinct vertices $t_i, t_j \in H_v$, if the gadget H contains p parallel t_it_j edges, then replace these edges by the following graph: Take a clique K_{k+1}, select $2p$ distinct vertices of it, and connect p of them to t_i, the other p to t_j, and delete a matching of size p induced by these $2p$ vertices from the clique. The graph substituting the parallel edges is connected, the degree of the original vertices of the gadgets do not change, and every newly added vertex has degree k. We show that an optimal order for this modified graph has as many vertices with left degree k as an optimal order for the original graph (which contains parallel edges): Consider an optimal order for the modified graph and delete the newly added vertices, this clearly does not increase the number of vertices with left degree k. To see the reverse direction, consider an optimal order for the original graph and insert each newly added vertex between the corresponding vertices t_i and t_j (assume that t_i precedes t_j, otherwise, change the role of t_i and t_j): First the vertices incident to t_i, after them the remaining vertices that are not incident to t_j and then the vertices incident to t_j. In the resulting order no newly added vertex has left degree equal to k and the degrees of the original vertices do not change.

This implies that the problem of finding a k-bounded order minimizing the vertices with left degree k is equivalent for the modified graph and for the original graph. □

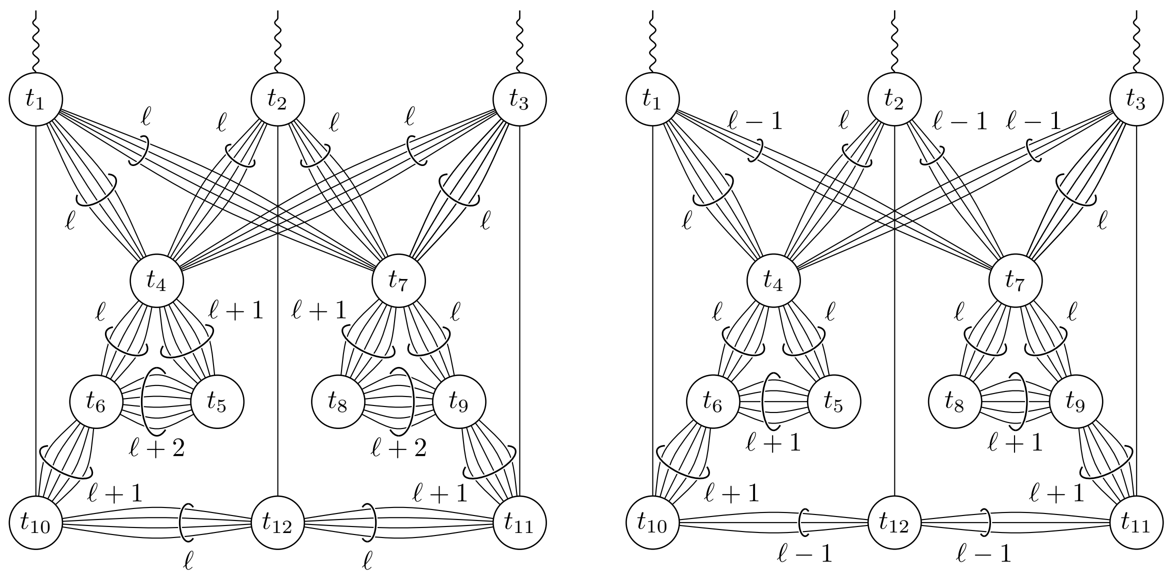
Since the dec-min k-bounded orders minimize the number of vertices with left degree exactly k, the previous theorem implies the following:

Corollary 21 *For any integer $k \geq 3$, finding a dec-min k-bounded order is NP-hard, even in case of simple graphs.*

Next, we prove the NP-hardness of the inc-max k-bounded ordering problem for any $k \geq 3$, and after that, the NP-hardness of the inc-max problem in the unbounded case.

Theorem 22 *For any integer $k \geq 3$, it is NP-hard to find a k-bounded order that minimizes the number of vertices with left outdegree exactly 0, even for simple graphs.*

Proof: First, we prove the theorem for multigraphs, then, we show how to eliminate the parallel edges. The proof is by a reduction from the vertex cover problem on 3-regular simple graphs [1], just as in the proof of Theorem 20.



(a) The case $k = 2\ell + 1$.

(b) The case $k = 2\ell$.

Figure 7: The gadgets corresponding to $v \in V$ for the proof of Theorem 22.

For a given 3-regular instance $G = (V, E)$ of the vertex cover problem, we construct the graph $G' = (V', E')$ the same way as in the proof of Theorem 20, the only difference is that we use the gadget H shown in Figure 7a if $k = 2\ell + 1$ for some $\ell \geq 1$, otherwise, if $k = 2\ell$ for some $\ell \geq 2$, then the one shown in Figure 7b. For every vertex v of G, extend G' with a disjoint copy H_v of the gadget H, and for every edge uv of G, add a new edge to G' between the vertices $t_1 \in H_u$, $t_2 \in H_u$ or $t_3 \in H_u$ and $t_1 \in H_v$, $t_2 \in H_v$ or $t_3 \in H_v$ in such a way that, in every gadget, exactly one of the three new edges is incident to the vertices t_1, t_2 and t_3. The wavy edges at the top of Figures 7a and 7b illustrate these new edges, which correspond to the edges in G, and will be referred to as original (wavy) edges.

We show that the minimum number of vertices with left degree 0 in the k-bounded orders of the vertices of G' equals the size of the smallest vertex cover in G plus the number of vertices in G. The construction of G' is slightly different for odd and even k, but the proof is given for the two cases simultaneously.

Firstly, let $X \subseteq V$ denote a minimum-size vertex cover in G. We construct a k-bounded order for G' in which the number of vertices with left degree 0 is $|X| + |V|$. We construct the order by removing vertices of degree at most k, and placing them to the last free position in the order of removal, from right to left. First, for each $v \in X$, remove the vertices of the gadget H_v in G' in the following order: $t_{12}, t_{11}, t_{10}, t_1, t_2, \ldots, t_9$. Then, for each $u \notin X$, remove the vertices of the gadget H_u in G' in the order $t_1, \ldots, t_{12}$. Clearly, the left degree of every vertex is at most k in the order obtained by this procedure. Notice that in the resulting order, for each $v \in X$, only the vertices $t_6 \in H_v$ and $t_9 \in H_v$ have left degree 0. Furthermore, for each $u \notin X$, only the vertex $t_{12} \in H_u$ has left degree 0. This means that the order found by the procedure above is a k-bounded order, and there are $|X| + |V|$ vertices with left degree 0, which was to be shown.

Secondly, given a k-bounded order for G', we construct a vertex cover $X \subseteq V$ of G whose size is at most $(\#\{v \in V : \overleftarrow{d}(v) = 0\} - |V|)$. Let the set $X \subseteq V$ contain a vertex $v \in V$ if and only if there exist at least two vertices in H_v whose left degree is 0 in the k-bounded order. We prove that X is a vertex cover by proving that for any $uv \in E$, if $u \notin X$, then u' precedes v' in the k-bounded order, where $u'v' \in E'$ is the only original (wavy) edge connecting H_u and H_v. Note that u' is one of $t_1, t_2, t_3 \in H_u$, and v' is one of $t_1, t_2, t_3 \in H_v$.

Suppose that v' precedes u' and consider the order of the vertices in the gadget H_u. Observe that in any k-bounded order, the last vertices of the 3-cycles $t_4t_5t_6$ and $t_7t_8t_9$ can only be t_4 and t_7, respectively, and these vertices must precede t_1, t_2 and t_3 (one of which is u'). If u' is t_2, then it has to precede t_{12}, otherwise, all of its neighbors would precede u'. Similarly, if u' is t_1 or t_3 then it has to precede t_{10} or t_{11}, respectively, which has to precede t_{12}. This means that t_1, t_2, t_3 and t_{12} all succeed the 3-cycles $t_4t_5t_6$ and $t_7t_8t_9$. This implies that the first 6 vertices of the gadget H_u span a subgraph of $H_u \setminus \{t_1, t_2, t_3, t_{12}\}$ which is not connected. The first vertices of the components have left degree 0 in any order, hence the gadget contains at least two vertices with left degree 0, contradicting $u \notin X$. This immediately implies that X is a vertex cover. Furthermore, in any k-bounded order, the first vertex of each gadget H_v has left degree 0, because only t_1, t_2 and t_3 have an incident edge going outside the gadget, and these vertices cannot be the first (otherwise, the last vertex of the 3-cycle $t_4t_5t_6$ would have left degree greater than k). Therefore, the size of X is at most $(\#\{v \in V : \overleftarrow{d}(v) = 0\} - |V|)$, which completes the proof of the theorem for multigraphs.

Now, we modify the construction so that the resulting graph becomes simple. For any $k \geq 3$, consider the corresponding gadget H_v shown in Figure 7. Let $\ell = \lfloor \frac{k}{2} \rfloor$. For any $p \in \{\ell - 1, \ell, \ell + 1, \ell + 2\}$ and any two distinct vertices t_i, t_j of H_v, if the gadget contains p parallel t_it_j edges, then replace these edges by the following graph: Take a complete bipartite graph $K_{k+1,k+1} = (L \cup R, E'')$, connect t_i with p distinct vertices from L and t_j with p distinct vertices from R, and delete a matching of size p induced by these $2p$ vertices from the complete bipartite graph. The graph substituting the parallel edges is connected, the degrees of the original vertices of the gadgets do not change, and every newly added vertex has degree $k+1$. So in every k-bounded order, the last vertex is either t_i or t_j. Furthermore, we can insert the newly added vertices between t_i and t_j into any order of the original vertices of the gadget without introducing any additional vertices with left degree 0. By symmetry, we can assume that t_j is later in the order than t_i is. Insert the newly added vertices after t_i and before t_j in the following order: first the vertices adjacent to t_i, then the vertices from R not adjacent to v, then the remaining vertices from L and finally, the remaining vertices from R. Therefore, from any order for the multigraph, we can construct an order for the modified graph without changing the number of vertices with left degree 0.

To see the reverse direction, consider an order given for the modified graph, and delete the newly added vertices. Observe that this modification does not increase the number of vertices with left degree 0, because consider an edge t_it_j where the degree of t_i decreased to 0. This means that t_i precedes t_j and at least one newly added vertex corresponding to the (parallel) edge t_it_j precedes t_i in the order. The first one of these newly added vertices has left degree 0. So for every vertex t_i whose left degree decreased to 0 we had to remove at least one corresponding newly added vertex. Therefore, an optimal order for this modified graph has at most as many vertices with left degree 0 as an optimal order for the multigraph, which completes the proof for simple graphs. □

Since an inc-max k-bounded order minimizes the number of vertices with left degree 0, the previous theorem implies the following.

Corollary 23 *For any integer $k \geq 3$, finding an inc-max k-bounded order is NP-hard, even in case of simple graphs.*

Next, we prove that finding an inc-max problem without bounds on the left degrees is NP-hard as well. Note that unlike Theorem 20 for dec-min orders, Theorem 22 does not imply the complexity of the inc-max problem without bounds on the left degrees, because it can happen that a graph has a k-bounded order, but the optimal order for the inc-max problem is not k-bounded, see the graph shown in Figure 5 for an example.

Theorem 24 *It is NP-hard to find an inc-max order, even for simple graphs.*

PROOF: Notice that an inc-max order for a simple graph maximizes the number ℓ for which there is exactly one vertex with left degree i for each $i \in \{0, \ldots, \ell - 1\}$. We prove that finding such an order is already NP-hard by a reduction from the MAX-3-SAT(4) problem, in which we are given a CNF formula such that each variable is contained in exactly 4 clauses, and each clause contains exactly 3 literals corresponding to 3 distinct variables. Our goal is to find an assignment that maximizes the number of satisfied clauses. This problem is APX-hard with an approximation threshold of $\frac{1900}{1899}$, that is, the problem cannot be approximated better than this threshold unless P=NP [4]. We show that a polynomial-time algorithm for determining the maximum number ℓ for which there exists an order with exactly one vertex with left degree i for each $i \in \{0, \dots, \ell-1\}$ would imply a better approximation ratio for the MAX-3-SAT(4) problem, hence the former problem is NP-hard, and so is finding an inc-max order.

For an instance of the MAX-3-SAT(4) problem with m clauses, we construct a graph $G=(V,E)$ as follows. For each $j \in \{1, \dots, m\}$ and $r \in \{1,2,3\}$, let G contain a vertex c_r^j corresponding to the r^{th} literal of the j^{th} clause. Add an edge between the vertices $c_{r_1}^{j_1}$ and $c_{r_2}^{j_2}$ if $j_1 \neq j_2$ and the corresponding literals are not each other's negations.

Let q denote the maximum number of clauses that can be satisfied by an assignment in the instance of the MAX-3-SAT(4) problem, and let ℓ denote the maximum integer such that there exists an order for G in which there is exactly one vertex with left degree equal to i for each $i \in \{1, \dots, \ell-1\}$. We prove that $q-4 \leq \ell \leq q$.

Firstly, consider the inequality $\ell \leq q$. Suppose that we are given an order of the vertices such that there is a unique vertex v_i of left degree i for each $i \in \{0, \dots, \ell-1\}$. Notice that each vertex in $V \setminus \{v_0, \dots, v_{\ell-1}\}$ has left degree at least ℓ, and G is simple, therefore, each such vertex must have at least ℓ preceding neighbors. This implies that the vertices $v_0, \dots, v_{\ell-1}$ are the first ℓ vertices in the order and they induce a clique. Therefore, any two of $v_0, \dots, v_{\ell-1}$ represent two literals that are not each other's negations, from two distinct clauses. So one can set the variables such that the literals corresponding to $v_0, \dots, v_{\ell-1}$ are true, and hence this assignment satisfies at least ℓ clauses, that is, $\ell \leq q$ holds.

Secondly, consider the inequality $q-4 \leq \ell$. Take an optimal solution to the instance of the MAX-3-SAT(4) problem that contains exactly q satisfied clauses. Choose a true literal from each of the satisfied clauses, and let $v_0, \dots, v_{q-1}$ denote the vertices corresponding to these literals. These vertices form a clique in G, because the corresponding literals are from different clauses and there are no two literals that are each other's negations. Consider an arbitrary order beginning with the vertices $v_0, \dots, v_{q-1}$. In this order, the left degrees of the vertices $v_0, \dots, v_{q-1}$ are $0, \dots, (q-1)$, respectively. All other vertices have left degree at least $(q-4)$, because each vertex $v \in V \setminus \{v_0, \dots, v_{q-1}\}$ has at least $(q-4)$ neighbors in $\{v_0, \dots, v_{q-1}\}$, since there is at most one non-neighbor corresponding to a literal from the same clause and 3 more corresponding to the negation of the literal corresponding to v. This means that there exists exactly one vertex with left degree i for $i \in \{0, \dots, (q-5)\}$, therefore, $q-4 \leq \ell$ holds.

By $q-4 \leq \ell \leq q$, if we can determine the value of ℓ, then we obtain a $\frac{q}{q-4}$-approximation for q. If $q \leq 7600$, then we can solve the MAX-3-SAT(4) problem in polynomial time, otherwise, if $q > 7600$, then $\frac{q}{q-4} > \frac{1900}{1899}$. So it is NP-hard to determine the value of ℓ, because one would obtain an approximation ratio better than $\frac{1900}{1899}$ for the MAX-3-SAT(4) problem, which implies that P=NP. □

3.2 The inc-min and the dec-max (acyclic) orientation problems

We have seen that both the dec-min and inc-max acyclic orientation problems are NP-hard. One can also define two natural counterparts, the inc-min and dec-max acyclic orientation problems: An acyclic orientation is optimal for the inc-min (increasingly minimal) problem if the sequence of the indegrees sorted in non-decreasing order is lexicographically minimal. Similarly, an acyclic orientation is optimal for the dec-max (decreasingly maximal) problem if the sequence of the indegrees sorted in non-increasing order is lexicographically maximal. We prove that these problems are NP-hard regardless of whether we require that the orientation is acyclic. The non-acyclic counterparts of these problems are referred to as the inc-min and the dec-max orientation problems.

Theorem 25 *The inc-min orientation and inc-min acyclic orientation problems are NP-hard, even for 3-regular simple graphs.*

Proof: Observe that the number of vertices of indegree zero is maximized in an optimal inc-min orientation and also in an inc-min acyclic orientation. Let n_0 and n_0^{DAG} denote the maximum number of zero indegrees in the former and latter problems, respectively. Clearly, $n_0 \geq n_0^{\mathrm{DAG}}$ holds.

We prove the NP-hardness of both problems simultaneously by a reduction from the maximum independent set problem, which is NP-hard even for 3-regular simple graphs [1]. Let us be given a 3-regular graph $G = (V, E)$ in which we want to determine the size of the maximum independent set, denoted by k. We show that $n_0 = n_0^{\mathrm{DAG}} = k$ holds, which immediately implies the NP-hardness of the inc-min orientation and inc-min acyclic orientation problems.

First, we prove that $n_0^{\mathrm{DAG}} \geq k$ holds. For an independent set of size k in G, consider the following acyclic orientation of G. Take a vertex order with the independent vertices in the first k places and the other vertices in the remaining places, and orient every edge from left to right. This is clearly an acyclic orientation, in which the vertices of the independent set have indegree 0.

Second, observe that $k \geq n_0$ holds, because the vertices of indegree zero are independent in any (not necessarily acyclic) orientation of G.

In summary, the following inequalities hold: $n_0 \geq n_0^{\mathrm{DAG}}$, $n_0^{\mathrm{DAG}} \geq k$ and $k \geq n_0$. These together imply that $n_0 = n_0^{\mathrm{DAG}} = k$ holds, which completes the proof. □

In the case of 3-regular graphs, an orientation is an inc-min acyclic orientation if and only if the reverse orientation is a dec-max acyclic orientation. To see this, observe that reversing an orientation preserves the acyclicity, and if the indegree of v is $\varrho(v)$ in an orientation, then the indegree of v is $(3 - \varrho(v))$ in the reversed orientation. The same argument holds for the analogous problems without acyclicity. So Theorem 25 implies the following corollary.

Corollary 26 *The dec-max orientation and dec-max acyclic orientation problems are NP-hard, even for 3-regular simple graphs.*

Table 1 summarizes the complexities of the considered orientation and acyclic orientation problems:

	dec-min	inc-max	inc-min	dec-max
orientation	P [6]	P [6, 13]	NP-H Thm 25	NP-H Cor 26
acyclic orientation	NP-H [6], Cor 21	NP-H Thm 24	NP-H Thm 25	NP-H Cor 26

Table 1: The complexities of the different lexicographical orientation problems without indegree bound.

4 Minimizing $\sum_{v \in V} h(\varrho(v))$ over acyclic orientations

This section studies the problem of finding an acyclic orientation for a given graph $G = (V, E)$ which minimizes $\sum_{v \in V} h(\varrho(v))$ for some discrete strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$ (i.e. $h(z + 2) + h(z) > 2h(z + 1)$ holds for any $z \in \mathbb{Z}_+$). András Frank and Kazuo Murota investigated the analogous question without acyclicity, that is, finding an orientation of G that minimize $\sum_{v \in V} h(\varrho(v))$. They showed that the exact same orientations are optimal for any discrete strictly convex function h, furthermore, these are also optimal for the (non-acyclic) dec-min and inc-max orientation problems [13]. Moreover, the optimal orientations can be found in strongly-polynomial time [6, 12].

Under the acyclicity constraint, however, the optimal solutions to different discrete strictly convex functions do not coincide anymore. We prove that the problem becomes NP-hard for *any* discrete strictly convex function h if we require acyclicity — provided that parallel edges are also allowed. After that, we give an exact dynamic programming algorithm for a generalization of the problem, and further examine the special case when $h(z) = z^2$, that is, when our goal is to find an acyclic orientation of G that minimizes the square-sum of the indegrees. Observe that finding an acyclic orientation of G minimizing $\sum_{v \in V} h(\varrho(v))$ is essentially the same as finding an order of the vertices minimizing $\sum_{v \in V} h(\overleftarrow{d}(v))$ (and orienting each edge from left to right). Throughout this section we are working with the ordering problems.

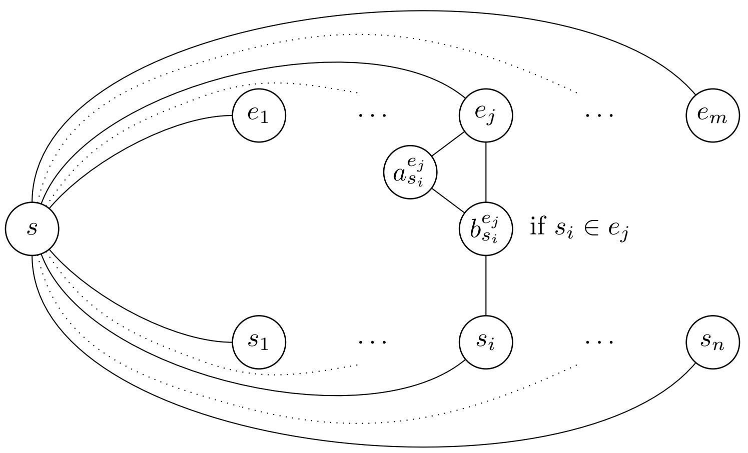


Figure 8: Illustration of the reduction in the proof of Theorem 28.

4.1 Hardness results

The goal of this section is to prove the following theorem about the complexity of finding a vertex order which minimizes $\sum_{v \in V} h(\overleftarrow{d}(v))$ for a discrete strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$.

Theorem 27 *Let us be given a multigraph without loops. It is NP-hard to find an order of the vertices minimizing $\sum_{v \in V} h(\overleftarrow{d}(v))$ for any discrete strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$.*

First, we prove the analogous statement for multigraphs in which loops are also allowed. Then, we use this weaker statement for proving Theorem 27. It is important to clarify the definition of $\overleftarrow{d}(v)$ in the case of loops: each loop incident to v contributes to $\overleftarrow{d}(v)$ by 1 in any order of the vertices. Note that a loop is a cycle of length one, hence no acyclic orientations exist in a graph with loops — but ordering the vertices is still possible.

Theorem 28 *Let us be given a multigraph $G = (V, E)$ in which loops are allowed. It is NP-hard to find an order minimizing $\sum_{v \in V} h(\overleftarrow{d}(v))$ for any discrete strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$.*

Proof: The proof is by a reduction from the set cover problem, which is NP-complete [17]. Let us be given an instance of the set cover problem with ground set $S = \{s_1, \dots, s_n\}$ and subsets $\mathcal{H} = \{e_1, \dots, e_m\} \subseteq 2^S$. Our goal is to decide whether there exist k subsets $e_{j_1}, \dots, e_{j_k} \in \mathcal{H}$ such that $\cup_{i=1}^k e_{j_i} = S$. We assume without loss of generality that $\cup_{j=1}^m e_j = S$, each item s_i is contained in at least 2 subsets, $n \geq k$ and $m \geq k$. We construct the multigraph $G = (V, E)$ as follows: Let G contain a vertex e_j for $j \in \{1, \dots, m\}$ corresponding to the subset $e_j \in \mathcal{H}$, and a vertex s_i for $i \in \{1, \dots, n\}$ corresponding to the item $s_i \in S$. Moreover, let G contain two vertices $a_{s_i}^{e_j}$ and $b_{s_i}^{e_j}$ for every $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$ such that $s_i \in e_j$. Add a triangle on the nodes $a_{s_i}^{e_j}, b_{s_i}^{e_j}$ and e_j, and add a new edge between $b_{s_i}^{e_j}$ and s_i. Moreover, add a new vertex s to G, and connect s to e_j for $j \in \{1, \dots, m\}$ and with s_i for $i \in \{1, \dots, n\}$. Add loops to the vertices such that $d(e_j) = d + 2$, $d(s_i) = d + 2$, $d(a_{s_i}^{e_j}) = d + 2$, $d(b_{s_i}^{e_j}) = d + 3$ for each $j \in \{1, \dots, m\}$ and $i \in \{1, \dots, n\}$, and $d(s) = d + 1 + n + k$, for some integer d. Figure 8 illustrates the construction, except for the loops. Note that d is polynomial in the size of the set cover problem, and hence so is the size of G.

We prove that there exist subsets $e_{j_1}, \dots, e_{j_k} \in \mathcal{H}$ covering S if and only if the optimal order for $\min_{v \in V} h(\overleftarrow{d}(v))$ contains at most k vertices with left degree at least $d + 2$.

We need to introduce some notations. Let $\mathcal{L}_{n',m',d',k'}$ denote the set of those multisets L of the integers for which

- $\ell \geq 0$ for each $\ell \in L$,
- $|L| = n'$,
- $\sum_{\ell \in L} \ell = m'$, where $m' = n'd' + r$ for some $r \in \{2k', \dots, n'\}$, and
- $\gamma_L \geq k'$, where $\gamma_L = \sum_{\ell \in L} [\ell - (d'+1)]^+$.

Let $L^* \in \mathcal{L}_{n',m',d',k'}$ denote the multiset that contains $(d'+2)$ with multiplicity k' and all other items are either $(d'+1)$ or d'. This means that L^* contains $(d'+2)$, $(d'+1)$ and d', with multiplicities k', $(r-2k')$ and $(n'+k'-r)$, respectively. The following claim states that L^* is the unique minimizer of $\sum_{\ell \in L} h(\ell)$ in $\mathcal{L}_{n',m',d',k'}$ for any discrete strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$.

Claim 29 *For any $L \in \mathcal{L}_{n',m',d',k'} \setminus \{L^*\}$ and any discrete strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$, the inequality $\sum_{\ell^* \in L^*} h(\ell^*) < \sum_{\ell \in L} h(\ell)$ holds.*

Proof: We show that if $L \in \mathcal{L}_{n',m',d',k'} \setminus \{L^*\}$, then L can be modified in such a way that $\sum_{\ell \in L} h(\ell)$ strictly decreases and the resulting multiset is also in $\mathcal{L}_{n',m',d',k'}$.

Firstly, suppose that L contains an element that is smaller than d'. Then it also contains an element that is at least $d'+1$, because $m' \geq n'd'+1$. If $d'+1 \in L$, then we can decrease $d'+1$ by one and increase an element smaller than d' by one. This does not change γ_L and strictly decreases $\sum_{\ell \in L} h(\ell)$, because h is strictly convex. Otherwise, if $d'+1 \notin L$, then

$$m' - \gamma_L = \sum_{\ell \in L} (\ell - [\ell - (d'+1)]^+) = \sum_{\ell \in L} \min\{\ell, d'+1\} < \sum_{\ell \in L} (d' + \mathbb{1}_{\ell \geq d'+1})$$
$$= n'd' + \#\{\ell \in L : \ell \geq d'+1\} = n'd' + \#\{\ell \in L : \ell \geq d'+2\} \leq n'd' + \gamma_L,$$

where the first inequality is true because L contains an element smaller than d', the last equation holds, since $d'+1 \notin L$, and all the other equations and inequalities follow by definition and rearrangements of the expressions. This implies that $r = m' - n'd' < 2\gamma_L$ holds. Since $r \geq 2k'$, we get that $\gamma_L > k'$. Therefore, by decreasing any element that is at least $d'+2$ by one, and increasing an element that is smaller than d' by one, the resulting multiset remains in $\mathcal{L}_{n',m',d',k'}$, and $\sum_{\ell \in L} h(\ell)$ strictly decreases.

Secondly, if all element of L are at least d' and there exists an element in L that is larger than $d'+2$, then $d' \in L$, because $m' \leq n'(d'+1)$. If $\gamma_L > k'$, then we can decrease an element that is larger than $d'+2$ by one, and increase d' by one. This change strictly decreases $\sum_{\ell \in L} h(\ell)$, and the resulting multiset remains in $\mathcal{L}_{n',m',d',k'}$. Otherwise, $\gamma_L = k'$, and hence

$$n'd' + r = m' = n'd' + \gamma_L + \#\{\ell \in L : \ell = d'+1\} + \#\{\ell \in L : \ell \geq d'+2\} <$$
$$n'd' + 2\gamma_L + \#\{\ell \in L : \ell = d'+1\} = n'd' + 2k' + \#\{\ell \in L : \ell = d'+1\},$$

where the first two equations come from the definition of m', the inequality is strict, because there exists an element in L that is larger than $d'+2$, which implies that $\#\{\ell \in L : \ell \geq d'+2\} < \gamma_L$, and the last equation is true because $\gamma_L = k'$. This implies that $r < 2k' + \#\{\ell \in L : \ell = d'+1\}$ holds. Since $r \geq 2k'$, we get that $d'+1 \in L$. We can decrease an element larger than $d'+2$ by one, and increase $d'+1$ by one. This change strictly decreases $\sum_{\ell \in L} h(\ell)$, and the resulting multiset is in $\mathcal{L}_{n',m',d',k'}$.

Otherwise, $d' \leq \ell \leq d'+2$ holds for each $\ell \in L$. As long as $\gamma_L > k'$, we know that $d' \in L$, because $m' \leq n'(d'+1)$. We can increase d' by one and decrease $d'+2$ by one, which strictly decreases the sum.

If $L \neq L^*$, then we are in one of the three cases considered above, therefore, there exists a modification of L which strictly decreases $\sum_{\ell \in L} h(\ell)$, and the resulting multiset remains in $\mathcal{L}_{n',m',d',k'}$. This implies that L^* is the unique minimizer of $\sum_{\ell \in L} h(\ell)$ in $\mathcal{L}_{n',m',d',k'}$, which we had to prove. $\square$

We continue the proof of Theorem 28. Consider the multisets in $\mathcal{L}_{n',m',d',k'}$, with $n' = |V|, m' = |E|$, $d' = d$ and $k' = k$ for the construction above, illustrated by Figure 8. Claim 29 states that there exists a unique minimizer L^* of $\sum_{\ell \in L} h(\ell)$ in $\mathcal{L}_{n',m',d',k'}$. Let opt_{L^*} denote the optimum value.

Claim 30 *There exist $e_{j_1}, \ldots, e_{j_k} \in \mathcal{H}$ covering S if and only if there exists an order with*

$$\sum_{v \in V} h(\overleftarrow{d}(v)) \leq opt_{L^*}.$$

Proof: First, suppose that there exist subsets $e_{j_1}, \ldots, e_{j_k} \in \mathcal{H}$ covering S. We show an appropriate order of the vertices of G. For $i = 1, \ldots, k$, repeat the following: put the vertex e_{j_i} to the last free position, delete it from G and while there exists a vertex of degree at most $d+1$, put that vertex to the last free position, and delete it from G. After that, put s to the last free position, then complete the order with the remaining vertices by applying the same method for e_j, where $j \notin \{j_1, \ldots, j_k\}$: put the vertex e_j to the last free position, delete it from G and while there exists a vertex of degree at most $d+1$ put that vertex to the last free position, and delete it from G. In the resulting order, the left degree of each vertex is d, $d+1$ or $d+2$, and exactly the vertices $e_{j_1}, \ldots, e_{j_k}$ have degree $d+2$ by the construction of the graph G. So the multiset consisting of the left degrees $\overleftarrow{d}(v)$ for $v \in V$ is exactly L^*, therefore, we get that $\sum_{v \in V} h(\overleftarrow{d}(v)) = \mathrm{opt}_{L^*}$, which proves the first direction of the statement.

Second, suppose that there exists an order with $\sum_{v \in V} h(\overleftarrow{d}(v)) \leq \mathrm{opt}_{L^*}$. Let L denote the multiset consisting of the left degrees $\overleftarrow{d}(v)$ for $v \in V$. Then $\gamma_L \leq k$, because $\gamma_L > k$ would imply that $L \in \mathcal{L}_{n',m',d',k'} \setminus \{L^*\}$ and $\sum_{\ell \in L} h(\ell) = \sum_{\ell^* \in L^*} h(\ell^*)$, which would contradict Claim 29.

Let us denote the number of the vertices e_j on the right side of s by ϵ_e, and the number of the vertices s_i on the right side of s by ϵ_s. Using these notations, we get that

$$k \geq \gamma_L \geq [\overleftarrow{d}(s) - (d+1)]^+ + \sum_{v \in V \,:\, s \text{ precedes } v} [\overleftarrow{d}(v) - (d+1)]^+ \geq (n + k - \epsilon_e - \epsilon_s) + \epsilon_e = n + k - \epsilon_s,$$

where the second equation is true because $[\overleftarrow{d}(s) - (d+1)]^+ = (n + k - \epsilon_e - \epsilon_s)$ and for each vertex e_j which is after s there is at least one corresponding vertex v (i.e. $v \in \{e_j, a^{e_j}_{s_i}, b^{e_j}_{s_i}\}$ for some $i \in \{1, \ldots, n\}$) with $\overleftarrow{d}(v)$ at least $d+2$. This implies that $\epsilon_s \geq n$ holds, so all the vertices $s_1, \ldots, s_n$ must be after s. Out of the (at most k) vertices with left degree at least $d+2$, we construct a set cover as follows. If $\overleftarrow{d}(e_j) = d+2$ or $\overleftarrow{d}(v) \geq d+2$ for $v \in \{a^{e_j}_{s_i}, b^{e_j}_{s_i}\}$ for some $i \in \{1, \ldots, n\}$, then include the subset e_j in the cover. If $\overleftarrow{d}(s_i) = d+2$, then include any subset $e_j \in \mathcal{H}$ containing s_i. What is left is to show that these subsets cover the whole ground set S. If $\overleftarrow{d}(s_i) = d+2$, then s_i is clearly covered. Otherwise, $\overleftarrow{d}(s_i) < d+2$, and then there exists a vertex $v \in \{e_j, a^{e_j}_{s_i}, b^{e_j}_{s_i}\}$ for some $j \in \{1, \ldots, m\}$ with $\overleftarrow{d}(v) \geq d+2$ by the construction of G, hence the subset $e_j \in \mathcal{H}$ corresponding to v covers s_i. □

Claim 30 completes the proof of Theorem 28 □

Now we turn to the proof of Theorem 27.

Proof: In the previous proof, we showed that it is NP-hard to minimize $\sum_{v \in V} h(\overleftarrow{d}(v))$. Consider the multigraph $G = (V, E)$ constructed in the proof (Figure 8 illustrates G except for the loops). We construct a loop-free multigraphs for which an optimal solution also has a part that is optimal for G. The construction consists of two steps.

First, construct a graph $G' = (V', E')$ as follows. Take a copy of G, add a new vertex u, replace each loop on v with an edge uv and add $d + 1 + n + k$ loops to u. We prove that u can be assumed to be the first vertex in an optimal order for G'. Consider any order in which some of its neighbors precede u and let v be the closest preceding neighbor of u. Observe that $\overleftarrow{d}(v) \leq d + 1 + n + k < d + 2 + n + k \leq \overleftarrow{d}(u)$ hold, therefore, moving u directly before v in the order does not increase the objective value. So we can assume that no neighbor of u precedes u. We can also assume without loss of generality that the rest of the nodes do not precede u either, because moving u to the front does not change the objective value. Consider an optimal order for G' in which u is the first vertex, and let σ denote the order of the other vertices. Then, $\overleftarrow{d}_{(u,\sigma)}(v') = \overleftarrow{d}_\sigma(v)$ holds for each $v' \in V' - u$ and $v \in V$, where $v' \in V'$ is the copy of $v \in V$. Therefore, the problem is equivalent for G and G' in the sense that an order (u, σ) is optimal for G' if and only if σ is optimal for G.

Second, construct a loop-free multigraph $G'' = (V'', E'')$ as follows. Take two disjoint copies $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ of $G' = (V', E')$ and denote the copy of $v \in V'$ by v_1 and v_2, in the two copies respectively. Remove the $d + 1 + n + k$ loops from both u_1 and u_2, and add $d + 1 + n + k$ parallel u_1u_2 edges between them. Now there are no loops in G''. Consider an optimal order for G''. Without loss of generality, assume that u_1 precedes u_2 (otherwise, switch the role of G_1 and G_2). This order restricted to V_2 is clearly an optimal solution to G', otherwise, we could decrease the objective value in G'' by changing the order of the vertices of V_2 in G'' to an optimal order of G'.

So from an optimal order for the loop-free multigraph G'', we can get an optimal order for G (by taking the order restricted to G_1 or G_2 and deleting u), which implies that the problem is NP-hard even for loop-free multigraphs, and hence completes the proof of Theorem 27. □

In Section 2.4 we gave two different characterizations for the convex hull of the indegree vectors of acyclic orientations. The NP-hardness of finding an acyclic orientation minimizing $\sum_{v\in V} h(\varrho(v))$ together with the characterization in Theorem 14 implies that finding a vector of the base-polyhedron $B(e_G)$ minimizing $\sum_{v\in V} h(\varrho(v))$ becomes NP-hard if we search for a corner solution.

Remark 31 *Note that if the multigraph $G = (V, E)$ has n vertices, then for $h(z) = n^z$, the optimal orders are exactly the dec-min orders, and for $h = (\frac{1}{n^z})$ the optimal orders are exactly the inc-max order. Therefore, Theorem 27 implies the NP-hardness of the dec-min and inc-max ordering problems in case of loop-free multigraphs. In Section 3.1, we already showed the hardness of the dec-min and inc-max problems even for simple graphs and in case of k-bounded orders.*

From the previous remark and Claim 18, it follows that the optimal solutions for $h(z) = n^z$ and $h = (\frac{1}{n^z})$ are not the same. This already shows that the optimal acyclic orientations for different discrete convex functions may not coincide, unlike the optimal orientations for the analogous orientation problem without the acyclicity condition [12, 13].

The strict convexity of the function h is a necessary condition in Theorem 27, because the problem is polynomial-time solvable in case of linear functions. We show this for a more general problem in which, instead of a single function h, each vertex $v \in V$ has its own linear function h_v. The solvability of this problem also follows of Theorem 14 and Theorem 15.

Theorem 32 *Let us be given a loop-free multigraph $G = (V, E)$ and for each $v \in V$, a discrete linear function $h_v : \mathbb{Z}_+ \to \mathbb{R}$. Ordering the vertices in non-increasing order by the slope of h_v minimizes $\sum_{v\in V} h_v(\overleftarrow{d}(v))$.*

Proof: Let a_v and b_v be coefficients such that $h_v(z) = a_v z + b_v$ for every $v \in V$. Consider an arbitrary order σ of the vertices, and let $\pi_\sigma(v) = b_v$ for every $v \in V$. Let $\pi_\sigma(uv) = a_v$ if u precedes v in σ, and otherwise a_u. Using this notation,

$$\sum_{v\in V} h_v(\overleftarrow{d}(v)) = \sum_{v\in V} a_v\overleftarrow{d}(v) + b_v = \sum_{e\in E} \pi_\sigma(e) + \sum_{v\in V} \pi_\sigma(v)$$

holds. Observe that in any order, $\pi(e) \geq \min\{a_u, a_v\}$ and $\pi(v) = b_v$. So we obtain an optimal solution by ordering the vertices in non-increasing order by a_v, because then $\pi(e) = \min\{a_u, a_v\}$. □

From the proof it also follows that if each vertex has the same linear function h, then any vertex order minimizes $\sum_{v\in V} h(\overleftarrow{d}(v))$.

4.2 A special case: Minimizing $\sum_{v\in V} \varrho^2(v)$ over acyclic orientations

In this section, we focus on the problem of finding an acyclic orientation minimizing the square-sum of the indegrees, which is essentially the same as finding an order of the vertices minimizing $\sum_{v\in V} \overleftarrow{d}^2(v)$. This problem is one of the most natural special cases of minimizing $\sum_{v\in V} h(\overleftarrow{d}(v))$, which we obtain by setting $h(z) = z^2$. The NP-hardness of the problem for loop-free multigraphs is immediate by Theorem 27.

In the next section, we strengthen this result by proving that the problem is hard for simple graphs as well. Then we propose a greedy algorithm and analyze its approximation ratio.

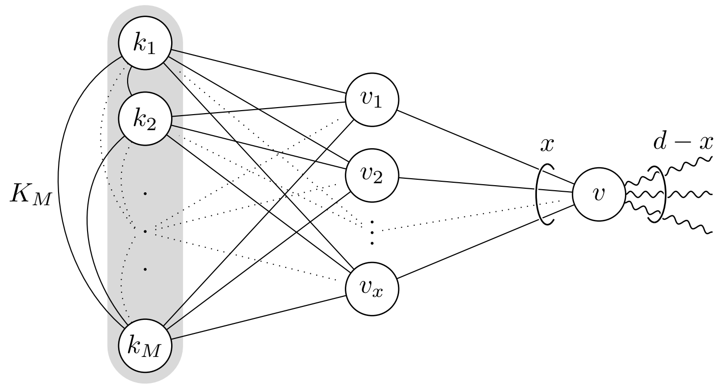


Figure 9: The gadget H_v for replacing x loops incident to the vertex v of total degree d in the multigraph constructed in the proof of Theorem 28. The wavy edges on the right illustrate the original edges.

4.2.1 Hardness for simple graphs

This section proves that finding an acyclic orientation that minimizes the square-sum of the indegrees is NP-hard even for simple graphs.

Theorem 33 *It is NP-hard to find a vertex order minimizing $\sum_{v\in V} \overleftarrow{d}^2(v)$, even for simple graphs.*

Proof: Consider the multigraph $G=(V,E)$ constructed in the proof of Theorem 28, which contains parallel edges. We assume without loss of generality that each vertex has at least 2 loops, because the number of loops on the vertices can be easily increased by increasing the parameter d. Start the construction of $G'=(V',E')$ by a copy of the graph G. For each vertex v with d incident edges, x of which are loops, remove all x incident loops, and add the gadget H_v which is constructed as follows. Take a clique K_M for $M=2d$, and denote its vertices by $k_1,\dots,k_M$. Add x new vertices $v_1,\dots,v_x$ to the gadget. Connect the vertices $v_1,\dots,v_x$ with v, and also add a complete bipartite graph between the vertices $v_1,\dots,v_x$ and the K_M. Figure 9 illustrates the construction.

Claim 34 *There exists an order for G' minimizing $\sum_{v\in V} \overleftarrow{d}^2(v)$ such that, for any vertex $v\in V$, the vertices of the gadget H_v are right before v in the order $k_1,\dots,k_M,v_1,\dots,v_x,v$, moreover, v is the last vertex of the gadget H_v in any optimal order.*

Proof: Let σ denote an optimal order that minimizes the total number of inversions against the order of the gadgets in the statement of the claim, that is, the number of those vertex pairs $u_1,u_2\in H_v$ for which u_1 precedes u_2 in σ and u_2 precedes u_1 in the order $k_1,\dots,k_M,v_1,\dots,v_x,v$ of the gadget H_v for some $v\in V$. We prove that the number of inversions is zero. On the contrary, suppose that there is a vertex $v\in V$ such that two vertices of H_v appear in reverse order in σ and in $k_1,\dots,k_M,v_1,\dots,v_x,v$. Observe that the left degree of each vertex in H_v-v only depends on the relative order of the vertices in H_v, therefore, we can assume that the vertices of H_v form an interval in σ. The vertices $k_1,\dots,k_M$ are symmetric, hence we can assume that they are in increasing order by the indices, otherwise, we could switch them into increasing order in σ, which leaves the order optimal, but decreases the number of inversions. For a similar reason, the vertices $v_1,\dots,v_x$ are also in increasing order. First, we focus on the order of the vertices in H_v-v_1 and we switch v_1 to the right place in the order at the end of the proof. Now suppose that there are some vertices of K_M after v_2 and take the smallest index j such that k_j is somewhere after v_2. Consider the vertex u right before k_j, which can only be v or v_i for some $i\in\{2,\dots,x\}$. If $u=v$, then there is no edge between v and k_j, so we can swap v and k_j without increasing $\sum_{v\in V} \overleftarrow{d}^2(v)$. The resulting order is also optimal and has fewer inversions than σ, which is a contradiction. If $u=v_i$ for some $i\in\{2,\dots,x\}$, then $\overleftarrow{d}(k_j)\ge j+1>j\ge \overleftarrow{d}(v_i)$. This means that we can swap v_i and k_j without increasing $\sum_{v\in V} \overleftarrow{d}^2(v)$. The resulting order is also optimal and has fewer inversions than σ, which is a contradiction. So the vertices of $H_v - v$ appear in σ in the order $k_1, \dots, k_{j-1}, v_1, k_j, \dots, k_M, v_2, \dots, v_x$ for some $j \in \{1, \dots, M+1\}$.

If v is after v_1 and before v_x, then by swapping v with the directly succeeding vertex w, the sum $\sum_{v\in V} \overleftarrow{d}^2(v)$ cannot increase. If w is a vertex in K_M, then the left degrees remain the same, therefore the resulting order is also optimal and has fewer inversions, which is a contradiction. Otherwise, if $w = v_i$ for some $i \in \{2, \dots, x\}$ then $\overleftarrow{d}(v)$ increases by one and $\overleftarrow{d}(w)$ decreases by one. Notice that $\overleftarrow{d}(v) \le d - 1$ and $\overleftarrow{d}(w) = M + 1$ before the change. Since $d \le M$, the sum strictly decreases if we swap v and w, which contradicts the optimality of σ.

Otherwise, if v and exactly k vertices of K_M precede v_1, then the following left degrees change if we put v after v_x: $\overleftarrow{d}(v) = a$ increases by x, $\overleftarrow{d}(v_1)$ decreases from $k+1$ to k, and $\overleftarrow{d}(v_i)$ decreases from $M+1$ to M for each $i \in \{1, \dots, x\}$. Hence the sum $\sum_{v\in V} \overleftarrow{d}^2(v)$ decreases by

$$\begin{aligned} &a^2 + (k+1)^2 + (x-1)(M+1)^2 - ((a+x)^2 + k^2 + (x-1)M^2) \\ &\qquad = -2ax + x^2 + 2k + 1 + (x-1)(2M+1) > -2dx + 2Mx - 2M. \end{aligned}$$

So it is enough to show that $(2M - 2d)x - 2M \ge 0$, which is true because $M = 2d$ and $x \ge 2$. The sum strictly decreases, which is a contradiction, because σ was optimal. This proves that the vertex v is the last among the vertices of H_v in σ. Moreover, the vertices of the gadget H_v must appear in σ in the order $k_1, \dots, k_{j-1}, v_1, k_j, \dots, k_M, v_2, \dots, v_x, v$ for some $j \in \{1, \dots, M+1\}$. If $j \ne M$, then we can swap v_1 and k_j without increasing the sum, and the number of inversions would decrease, which is a contradiction. This means that the number of inversions is zero, and hence the vertices of the gadget H_v appear in σ in the order $k_1, \dots, k_M, v_1, v_2, \dots, v_x, v$, which completes the proof of the claim. □

We continue the proof of Theorem 33. Let σ_{H_v} denote the order $k_1, \dots, k_M, v_1, \dots, v_n, v$, and let $\mathrm{opt}_{H_v} = \sum_{u \in H_v - v} \overleftarrow{d}_\sigma(u)$. Let opt_G and $\mathrm{opt}_{G'}$ denote the optimum value for G and for G', respectively.

To finish the proof, it suffices to prove that $\mathrm{opt}_{G'} = \mathrm{opt}_G + \sum_{v\in V} \mathrm{opt}_{H_v}$ and if an order σ' is optimal for G' then the same order restricted to the vertices of V is optimal for G. Let σ denote this restricted order.

First, if an order σ is optimal for G, then insert the vertices of the gadget H_v right before v into σ in the order of σ_{H_v}. Then the resulting order has objective value $\mathrm{opt}_G + \sum_{v\in V} \mathrm{opt}_{H_v}$. From this, $\mathrm{opt}_{G'} \le \mathrm{opt}_G + \sum_{v\in V} \mathrm{opt}_{H_v}$ follows.

Second, let σ' be an optimal order for G'. By Claim 34, we can assume that for each $v \in V$, the vertices of H_v are right before v in the order $k_1, \dots, k_M, v_1 \dots, v_n$, otherwise, we can rearrange the vertices of H_v into this order. This implies that for each vertex $u \in V'$,

$$\overleftarrow{d}_{\sigma'}(u) = \begin{cases} \overleftarrow{d}_\sigma(u) & \text{if } u \in V, \\ \overleftarrow{d}_{\sigma_{H_v}}(u) & \text{if } u \in H_v - v \text{ for some } v \in V, \end{cases}$$

therefore, $\mathrm{opt}_{G'} = \sum_{v\in V} \overleftarrow{d}_\sigma(v) + \sum_{v\in V} \mathrm{opt}_{H_v} \ge \mathrm{opt}_G + \sum_{v\in V} \mathrm{opt}_{H_v}$.

These together imply that $\mathrm{opt}_{G'} = \mathrm{opt}_G + \sum_{v\in V} \mathrm{opt}_{H_v}$ and σ' restricted to V is optimal for G. This completes the proof, because finding an optimal order for G is NP-hard by Theorem 28. □

4.2.2 Greedy approximation algorithm

This section investigates the approximation ratio of a greedy algorithm for the problem of finding an order minimizing the square-sum of the left degrees, which achieves an approximation guarantee of $\min\{4\eta_n, \overleftarrow{d}_{\min}(G)\}$, where $\eta_n = \sum_{i=1}^n \frac{1}{i}$ is the n^{th} harmonic number and $\overleftarrow{d}_{\min}(G)$ is the degeneracy of the graph. Recall that the degeneracy of G is the same as the minimum number k for which G has a k-bounded order [20]

The algorithm repeats the following, until no vertex remains: It fixes a vertex with minimum degree at the last free position and deletes it from the graph.

Note that there can be multiple vertices with minimum degree, so the output of the algorithm depends on which minimum-degree vertex we choose.

Observe, that we already mentioned this algorithm in Section 2.2.2. It was first given in [21] for determining the degeneracy of G by computing a $\overleftarrow{d}_{\min}(G)$-bounded order. So the output of the greedy algorithm $v_1, \ldots, v_n$ is known to be a $\overleftarrow{d}_{\min}(G)$-bounded order. We will use this fact in the proof of Theorem 35.

In what follows, we prove two different upper bounds on the approximation ratio of this algorithm. The first one gives a better guarantee for sparse graphs and the other one for dense graphs.

Theorem 35 *Let us be given a graph $G = (V, E)$. The greedy algorithm gives a $\overleftarrow{d}_{\min}(G)$-approximate order for minimizing $\sum_{v \in V} \overleftarrow{d}^2(v)$.*

Proof: Consider the greedy order, and for each edge e, let $\pi(e) = \overleftarrow{d}(v)$, where v is the endpoint of e that is later in the order. Notice that $\sum_{e \in E} \pi(e) = \sum_{v \in V} \overleftarrow{d}^2(v)$. Consider any edge $e \in E$. Observe that $\pi(e) = \overleftarrow{d}(v)$ for some $v \in V$ and $\overleftarrow{d}(v) \le \overleftarrow{d}_{\min}(G)$, because the greedy algorithm computes a $\overleftarrow{d}_{\min}(G)$-bounded order, as we already mentioned. This implies that $\pi(e)$ in the greedy order is at most $\overleftarrow{d}_{\min}(G)$, therefore, the objective value of the greedy order is at most $|E|\overleftarrow{d}_{\min}(G)$. Since the optimum is at least $|E|$, this together implies that $\frac{\text{obj}}{\text{opt}} \le \frac{|E|\overleftarrow{d}_{\min}(G)}{|E|} = \overleftarrow{d}_{\min}(G)$. □

We give another upper bound for the approximation ratio.

Theorem 36 *Let us be given a graph $G = (V, E)$. The greedy algorithm gives a $4\eta_n$-approximate order for minimizing $\sum_{v \in V} \overleftarrow{d}^2(v)$, where $\eta_n = \sum_{i=1}^{n} \frac{1}{i}$ is the n^{th} harmonic number.*

Proof: Let $v_1, \ldots, v_n$ denote the order given by the greedy algorithm and let obj denote its objective value. We introduce the notation $V_i = \{v_1, \ldots, v_i\}$ for the set of the first i vertices, and consider the graph $G[V_i] = (V_i, E_i)$ induced by V_i. The size of E_i will be denoted by m_i, and we use $\text{opt}(V_i)$ to denote the optimum value for the graph $G[V_i]$.

Then the following lower bound holds for the optimum value for all $i \in \{1, \ldots, n\}$:

$$\text{opt}(V) \ge \text{opt}(V_i) \ge \sum_{v \in V_i} \left(\frac{m_i}{i}\right)^2 = \frac{m_i^2}{i}, \tag{2}$$

where the first inequality holds, because the optimal order contains an order of the vertices in V_i, and the second inequality follows by rearrangement.

The following upper bound holds for the objective value of the greedy order:

$$\text{obj} = \sum_{i=1}^{n} \overleftarrow{d}(v_i) \le \sum_{i=1}^{n} \left(\frac{2m_i}{i}\right)^2 = 4\sum_{i=1}^{n} \frac{1}{i}\frac{m_i^2}{i} \le 4\sum_{i=1}^{n} \frac{1}{i}\text{opt}(V) = 4\eta_n \text{opt}(V),$$

where the first inequality holds because v_i is a vertex with minimum degree in $G[V_i]$ and the average degree in $G[V_i]$ is $\frac{2m_i}{i}$. The second inequality uses lower bound which was derived in (2). □

The best lower bound that we found for the approximation ratio is $\frac{9}{7}$, which is a limit of the worst-case approximation ratios of the greedy algorithm for the sequence of graphs shown in Figure 10.

We describe a worst-case greedy order and then an optimal order for the graph shown in Figure 10 with k triangles. Consider the order which begins with the vertices v_i^1, v_i^2, v_i^3 for $i = 1, \ldots, k$ and after that contains the vertices u_i for $i = 2, \ldots, k$. Note that for every $\ell \in \{1, \ldots, n\}$, the ℓ^{th} vertex in this order has minimum degree in the graph induced by the first ℓ vertices. So this solution can be found by the greedy algorithm above, and the objective value of this order is $k(0^2 + 1^2 + 2^2) + (k-1)(2^2) = 9k - 4$. In fact, one can also argue that this is the worst possible solution that the greedy algorithm may return for G_k. On the other hand, the following order is optimal. List the vertices v_1^1, v_1^2, v_1^3

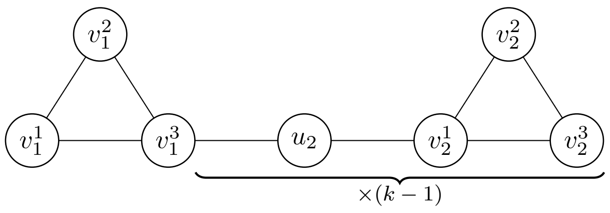


Figure 10: Illustration of the graph sequence G_k for which the approximation ratio of the greedy algorithm tends to $\frac{9}{7}$. Here k denotes the number of the triangles, and v_i^j denotes the ith vertex of the jth triangle, for each $i \in \{1,\dots,k\}$ and $j \in \{1,\dots,3\}$, and u_i denotes the vertex connecting v_{i-1}^3 and v_i^1 for each $i \in \{2,\dots,k\}$.

first, and after them the vertices u_i, v_i^1, v_i^2, v_i^3 in this order for $i = 2,\dots,k$. So the optimum value is $(0^2+1^1+2^2)+(k-1)(1^2+1^2+1^2+2^2) = 7k-2$. Therefore, the approximation ratio is $\frac{9k-4}{7k-2}$, which tends to $\frac{9}{7}$ as k goes to infinity.

We tested the approximation ratio of the worst possible solution found by the greedy algorithm for every simple graph on at most 12 vertices, and found that the approximation ratio for the graph G_k, shown in Figure 10, is an upper bound for the approximation ratio for every simple graph on at most $(4k-1)$ vertices. It is an open question whether the approximation ratio of the greedy algorithm is $\frac{9}{7}$ for simple graphs.

4.3 An exponential dynamic programming algorithm

In Section 4.1, we proved that it is NP-hard to find a vertex order minimizing $\sum_{v\in V} h(\overleftarrow{d}(v))$ for any discrete strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$ in case of loop-free multigraphs. In this section, we give an exact method for solving the following more general problem. Let us be given a multigraph $G = (V,E)$ and a discrete (not necessarily strictly convex) function $h_v : \mathbb{Z}_+ \to \mathbb{R}$ for each $v \in V$. Assume that the function h_v can be evaluated efficiently for each $v \in V$. Our goal is to find an order minimizing $\sum_{v\in V} h_v(\overleftarrow{d}(v))$. The natural approach to solve this problem is to try all $|V|!$ permutations of the vertices, and choose one minimizing the objective value.

We give a dynamic programming algorithm for finding an order minimizing $\sum_{v\in V} h_v(\overleftarrow{d}(v))$, which takes $O(2^{|V|}\text{poly}(|V|,|E|))$ steps. Let $f(\emptyset)=0$. For each $\emptyset \neq V' \subseteq V$, in non-decreasing order by $|V'|$, compute and memoize

$$f(V') = \min_{v\in V'}\{f(V'-v) + h_v(d(v,V'))\}, \tag{3}$$

and choose

$$g(V') \in \operatorname*{arg\,min}_{v\in V'}\{f(V'-v) + h_v(d(v,V'))\}. \tag{4}$$

After that, construct the optimal order by repeating the following step until no vertex remains: Put $g(V)$ at the last free place of the order, and delete v from the graph.

We prove the correctness of this algorithm by showing that $f(V')$ is the optimum value and $g(V')$ is the last vertex of an optimal order for $G[V']$ for each $V' \subseteq V$.

Theorem 37 *For each subset $\emptyset \neq V' \subseteq V$, the minimum value of $\sum_{v\in V'} h_v(\overleftarrow{d}(v))$ for the graph $G[V']$ is $f(V')$ defined in (3), and there exists an optimal order in which the last vertex is $g(V')$ defined in (4).*

Proof: The proof is by induction on $|V|$. Consider a graph $G=(V,E)$. Suppose that the statement holds for every $V' \subset V$.

Firstly, consider an optimal order $\sigma = v_1,\dots,v_n$ for G. Notice that $v_1,\dots,v_{n-1}$ is an optimal order for $G-v_n$. By induction, the objective value of this order is $f(V-v_n)$. Moreover, v_n is the last vertex in σ, therefore, $\overleftarrow{d}(v_n) = h_{v_n}(d(v_n,V))$ holds. These imply that $f(V-v_n)+h_{v_n}(d(v_n,V))$ is the objective value of σ. So $f(V)$ is at most the optimum value for G.

Secondly, construct an order for G with objective value $f(V)$. Let v_n denote a vertex, for which $f(V-v)+h_v(d(v,V))$ is minimal (so $v_n \in \arg\min_{v\in V'}\{f(V'-v)+h_v(d(v,V'))\}$). The inductive hypothesis implies that $f(V-v_n)$ is the optimum value for $G-v_n$. Take an optimal order for $G-v_n$ and put v_n at the end of this order. The resulting order has objective value equal to $f(V)$. This completes the proof. □

The running time of the dynamic programming algorithm is clearly $O(2^{|V|}\text{poly}(|V|,|E|))$, because we assumed that the h_v functions can be evaluated in polynomial time, so the computation of $f(V')$ and $g(V')$ takes $\text{poly}(|V|,|E|)$ time for each subset.

5 Maximizing $\sum_{v\in V}\varrho(v)\delta(v)$ over acyclic orientations

This section is devoted to another notion of "equitable" acyclic orientations, that is, we define the equitable orientations in terms of the in- and outdegrees as opposed to Problems 1, 2 and 3, which only focus on the indegrees.

Our goal is to find an acyclic orientation of a graph $G=(V,E)$ maximizing $\sum_{v\in V}\varrho(v)\delta(v)$. The optimal orientations are equitable in the sense that the product $\varrho(v)\delta(v)$ is maximal for a single vertex v if the degree of v is evenly distributed to the indegree and outdegree of v.

Without the acyclicity condition, the optimal orientations are either the Eulerian (that is, $\varrho(v)=\delta(v)$ for every $v\in V$) or the almost-Eulerian (that is, $|\varrho(v)-\delta(v)|\le 1$ for every $v\in V$) orientations. It is well known that every graph has an Eulerian or almost-Eulerian orientation and we can compute such orientations in polynomial time, hence the problem is polynomial-time solvable. However, the acyclicity condition makes the problem much more difficult, as we are going to see in the next section.

5.1 NP-hardness

The problem of finding an acyclic orientation of G which maximizes $\sum_{v\in V}\varrho(v)\delta(v)$ is clearly equivalent to the problem of finding an order of the vertices maximizing $\sum_{v\in V}\overleftarrow{d}(v)\vec{d}(v)$. In [5], the authors considered similar ordering problems in which they wanted to find an order minimizing $\sum_{v\in V}|\overleftarrow{d}(v)-\vec{d}(v)|$. They called an order *perfectly balanced* if $|\overleftarrow{d}(v)-\vec{d}(v)|\le 1$ holds for every vertex v, and proved that deciding whether a given graph has a perfectly balanced vertex order is NP-complete. Next, we describe their proof.

Theorem 38 (Biedl, Chan, Ganjali, Hajiaghayi, Wood [5]) *It is NP-complete to decide whether a graph has a perfectly balanced vertex order.*

Proof: The proof is by reduction from the positive NAE-(2,3)-SAT(3) problem, in which we are given a conjunctive normal form (CNF) formula which only contains positive literals, each clause contains 2 or 3 literals and each variable occurs at most 3 times. Our goal is to decide whether there exists a truth assignment for the variables such that each clause contains at least one true literal and at least one false literal. This problem is NP-complete [19].

Let us be given an instance of the positive NAE-(2,3)-SAT(3) problem with variables $x_1,\dots,x_n$ and clauses $c_1,\dots,c_m$, and denote the number of occurrences of the variable x_i by d_i. Construct the graph $G=(V,E)$ as follows. For each variable x_i for $i\in\{1,\dots,m\}$ which occurs $d_i\in\{1,2,3\}$ times in the CNF formula, add the gadget H_{x_i} containing a path on the vertices $v_0^i,\dots,v_{2d_i}^i$ and an edge $v_0^iv_{2\ell-1}^i$ for $j\in\{2,\dots,d_i\}$. Moreover, for each clause c_j, let G contain a vertex c_j and add an edge $c_jx_0^i$ for every variable x_i which is in c_j. The gadget H_{x_i} for a variable x_i with $d_i=3$ occurrences is shown in Figure 11. We claim that the given instance of the positive NAE-(2,3)-SAT(3) is satisfiable if and only if G has a perfectly balanced vertex order.

First, if the positive NAE-(2,3)-SAT(3) instance is satisfiable, then construct the vertex order as follows. Put the vertices corresponding to the clauses in the middle of the order continuously. For each variable x_i set to true, put the vertices of the gadget H_{x_i} before the vertices corresponding to the clauses

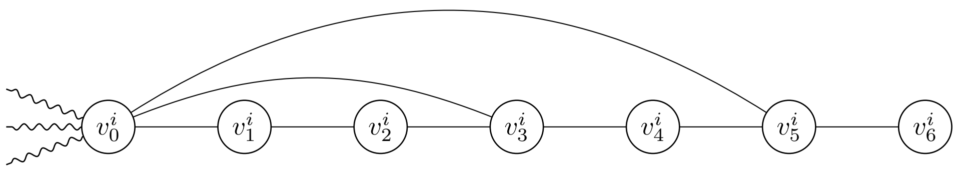


Figure 11: Example for the gadget H_{x_i} constructed in the proof of Theorem 38, for a variable x_i with $d_i = 3$ occurrences. The wavy edges are going to the vertices of the clauses containing x_i.

in order $v^i_{2d_i}, \dots, v^i_0$; and for each variable x_i set to false, put the vertices of the gadget H_{x_i} after the clause vertices in order $v^i_0, \dots, v^i_{2d_i}$. It is easy to see that the vertices $v^i_1, \dots, v^i_{d_i}$ from the gadget H_{x_i} are balanced in the obtained order for every $i \in \{1, \dots, n\}$. The vertex v^i_0 is also balanced for every $i \in \{1, \dots, n\}$, because it has d_i incident edges inside the gadget H_{x_i} which are going to the same direction, and d_i incident edges going to vertices corresponding to clauses which are going to the opposite direction. Each vertex corresponding to a clause has 2 or 3 incident edges, and at least one of these edges goes to the left, because the clause contains at least one true literal, similarly at least one edge goes to the right, because the clause also contains a false literal. Therefore, every vertex is perfectly balanced.

Second, if there exists a perfectly balanced order of the vertices, then consider the vertices of the gadget H_{x_i}. If v^i_1 is to the right (left) from v^i_0, then v^i_2 has to be to the right (left) from v^i_1, because v^i_1 is balanced. Similarly, for every $\ell \in \{1, \dots, d_i\}$, v^i_ℓ has to be to the right (left) from $v_{\ell-1}$, because $v_{\ell-1}$ is balanced. So the vertices of the gadget H_{x_i} are either in the order $v^i_{2d_i}, \dots, v^i_0$ or in the order $v^i_0, \dots, v^i_{2d_i}$. Set the variable x_i to true if the gadget H_{x_i} belongs to the former category, otherwise, set it to false. Since the vertex v^i_0 is balanced, every vertex corresponding to a clause containing x_i has to be to the right (left) of the vertex v^i_0, if x_i is true (false). Note that each vertex corresponding to a clause has degree 2 or 3, therefore, it has at least one edge going to the left and at least one edge going to the right. This means that each clause contains at least one true and at least one false literal, therefore, the constructed truth assignment satisfies the positive NAE-(2,3)-SAT(3) instance.

□

Note that the proof of Theorem 38 only implies the NP-completeness for graphs with maximum degree 6. In [16], the authors strengthened this theorem and proved that deciding whether a graph has a perfectly balanced vertex order is NP-hard even for planar graphs with maximum degree at most 4 and for 5-regular graphs.

For a single vertex v, the value of $\overleftarrow{d}(v)\overrightarrow{d}(v)$ is maximized if and only if $|\overleftarrow{d}(v) - \overrightarrow{d}(v)| \leq 1$. So if G has a perfectly balanced order, then exactly the perfectly balanced orders are maximizing $\sum_{v \in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$. Therefore, by the NP-hardness of finding a perfectly balanced order, we obtain the following.

Corollary 39 *It is NP-hard to find a vertex order maximizing $\sum_{v \in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$, even for graphs with maximum degree at most* 4.

In the next section, we prove that our problem is tracktable when the maximum degree is at most 3.

5.2 Polynomial-time algorithm for graphs with maximum degree at most 3

The problem of finding an order minimizing $\sum_{v \in V} |\overleftarrow{d}(v) - \overrightarrow{d}(v)|$ is polynomial-time solvable, for simple graphs with maximum degree at most 3. The algorithm given in [5] finds an order minimizing the number of vertices for which $\overleftarrow{d}(v) = 0$ or $\overrightarrow{d}(v) = 0$. We prove that, a slightly modified version of their algorithm solves our problem in case of graphs with maximum degree at most three. Before presenting the algorithm we need the following definitions. An order for $G = (V, E)$ is called an *s-t order* if s and t are the first and last vertices of the order, respectively, and $\overleftarrow{d}(v) \geq 1$ and $\overrightarrow{d}(v) \geq 1$ hold for each $v \in V \setminus \{s, t\}$ if $d(v) \geq 2$. Every biconnected graph has an s-t order for any distinct vertices $s, t \in V$, and we can compute such an order in polynomial time [10]. Clearly, an s-t order maximizes $\sum_{v \in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$, for any biconnected

3-regular graph, we extend this for any connected graph with maximum degree at most 3. Let $B_1, \ldots, B_r$ denote the biconnected components of G. It is well known that the biconnected components can be represented by a tree T. A component is called an end-component if it only contains one cut vertex (i.e. the end-components are the leaves of T). The following algorithm is a modified version of the algorithm presented in [5].

Algorithm 3 COMBINE s-t ORDERINGS

```
1: Let us be given a connected graph $G = (V, E)$.
2: Determine the tree $T$ of the biconnected components of $G$.
3: Let $B_1$ be an end-component.
4: Start a depth-first traversal of $T$ from $B_1$ and let $B_1, \ldots, B_r$ be the depth-first numbering of the
   biconnected components of $G$.
5: Let $t_1 \in B_1$ be the unique cut vertex of $B_1$ and choose $s_1 \in B_1 - t_1$ such that $d_G(s_1)$ is minimal. If
   no such vertex exists, then let $s_i = t_i$.
6: Compute an $s_1$-$t_1$ order for $B_1$, and denote this order by $\sigma^1$.
7: for $i = 2, \ldots, r$ do
8:     Let $s_i$ be a cut vertex of $B_i$ with a block $B_j$ for some $j < i$.
9:     if $B_i$ is an end-component of $G$ then
10:        Choose a vertex $t_i \in B_i - s_i$ such that $d_G(t_i)$ is minimal, if no such vertex exists, then let $t_i = s_i$.
11:    else
12:        Let $t_i$ be a cut vertex of $B_i$ with a component $B_j$ for some $j > i$.
13:    end if
14:    Let $v^i_1, \ldots, v^i_{n_i}$ be an $s_i$-$t_i$ order of $B_i$.
15:    Let $\sigma^i$ denote the order obtained by adding $v^i_2, \ldots, v^i_{n_i}$ to the end of $\sigma^{i-1}$.
16: end for
17: output $\sigma^r$
```

Algorithm 3 computes s-t orders for the biconnected components of G and combines them into a single order.

Theorem 40 *Let us be given a simple connected graph $G = (V, E)$, with maximum degree at most 3. Algorithm 3 computes an order maximizing $\sum_{v \in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$ in polynomial time.*

PROOF: The running time of the algorithm is clearly polynomial. We define the *imbalance* of v as

$$\mathcal{I}_\sigma(v) = \left\lfloor \frac{d(v)}{2} \right\rfloor \left\lceil \frac{d(v)}{2} \right\rceil - \overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v).$$

Note that $\mathcal{I}_\sigma(v) \geq 0$, because $\left\lfloor \frac{d(v)}{2} \right\rfloor \left\lceil \frac{d(v)}{2} \right\rceil$ is an upper bound for $\overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)$ in any order σ.

Clearly, the problem of maximizing $\sum_{v \in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$ is equivalent to finding an order minimizing $\sum_{v \in V} \mathcal{I}_\sigma(v)$. We prove the correctness of Algorithm 3 for the latter problem.

By definition,

$$\mathcal{I}_\sigma(v) = \begin{cases} 2 & \text{if } d(v) = 3 \text{ and } \overleftarrow{d}(v) = 0 \text{ or } \overrightarrow{d}(v) = 0, \\ 1 & \text{if } d(v) = 2 \text{ and } \overleftarrow{d}(v) = 0 \text{ or } \overrightarrow{d}(v) = 0, \\ 0 & \text{otherwise,} \end{cases}$$

in any order σ, which means that $\mathcal{I}_\sigma(v)$ is either $(d(v) - 1)$ or 0.

Let T denote the tree of the biconnected components of G and let $B_1, \ldots, B_r$ denote the biconnected components of G, numbered in the depth-first numbering as in Line 4 of the algorithm. Suppose that $B_{i_1}, \ldots, B_{i_\ell}$ are the end-components, where $B_{i_1} = B_1$. Let q_{i_j} denote the unique cut vertex of B_{i_j} for each $j \in \{1, \ldots, \ell\}$. So with the notations in Algorithm 3, $q_{i_1} = t_{i_1}$ and $q_{i_j} = s_{i_j}$ for each $j \in \{2, \ldots, \ell\}$.

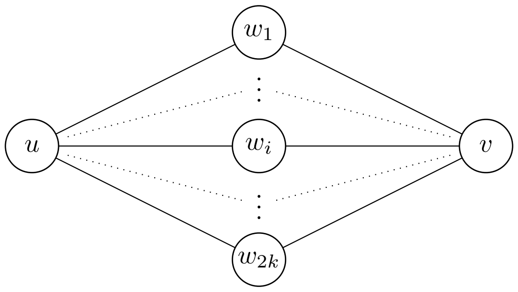


Figure 12: The graph sequence G_k which shows that the $\frac{13}{8}$-approximation algorithm for $\sum_{v\in v}\max\{\overleftarrow{d}(v),\overrightarrow{d}(v)\}$ in [5] does not approximate our problem within any constant factor.

Observe that in the order σ^r given by the algorithm, each $v\in V\setminus\{s_{i_1},t_{i_2},\dots,t_{i_\ell}\}$ have $\overleftarrow{d}_{\sigma^r}(v)\geq 1$ and $\overrightarrow{d}_{\sigma^r}(v)\geq 1$, therefore, $\mathcal{I}_{\sigma^r}(v)=0$ for these vertices. This implies that

$$\sum_{v\in V}\mathcal{I}_{\sigma^r}(v)=\mathcal{I}_{\sigma^r}(s_{i_1})+\sum_{j=2}^{\ell}\mathcal{I}_{\sigma^r}(t_{i_2})\leq d(s_{i_1})-1+\sum_{j=2}^{\ell}(d(t_{i_j})-1). \tag{5}$$

Consider any order σ of V. In a biconnected component B_{i_j}, only the vertex q_{i_j} is connected to vertices of other components, hence the first or last vertex of B_{i_j} in the order is some vertex $v_{i_j}\in B_{i_j}-q_{i_j}$, which implies that $\overleftarrow{d}_\sigma(v_{i_j})=0$ or $\overrightarrow{d}_\sigma(v_{i_j})=0$. Therefore, $\mathcal{I}_\sigma(v_{i_j})=d(v_{i_j})-1$ and

$$\sum_{v\in V}\mathcal{I}_\sigma(v)\geq\sum_{j=1}^{\ell}\mathcal{I}_\sigma(v_{i_j})=\sum_{j=1}^{\ell}(d(v_{i_j})-1). \tag{6}$$

Observe that $d(s_{i_1})\leq d(v_{i_1})$ holds, because $s_{i_1},v_{i_1}\in B_{i_1}-q_{i_1}$, and Algorithm 3 chose s_{i_1} in Line 5. Similarly, $d(t_{i_j})\leq d(v_{i_j})$ holds for each $j\in\{1,\dots,\ell\}$, because both $t_{i_1},v_{i_1}\in B_{i_1}-q_{i_1}$ and Algorithm 3 chose t_{i_1} in Line 10. These with the equations (5) and (6) together imply that $\sum_{v\in V}\mathcal{I}_{\sigma^r}(v)\leq\sum_{v\in V}\mathcal{I}_\sigma(v)$ holds, which implies that σ^r is an optimal solution. □

So the problem of finding an order maximizing $\sum_{v\in V}\overleftarrow{d}(v)\overrightarrow{d}(v)$ is polynomial-time solvable in case of simple graphs with maximum degree at most three, but becomes NP-hard for simple graphs with maximum degree at most four.

From now on, we focus on the problem of approximating $\max\sum_{v\in V}\overleftarrow{d}(v)\overrightarrow{d}(v)$. Note that, in [5], a $\frac{13}{8}$-approximation algorithm was given for the problem of finding an order minimizing $\sum_{v\in V}\max\{\overleftarrow{d}(v),\overrightarrow{d}(v)\}$. The algorithm constructs an order by inserting the vertices one by one, placing each in the middle of their already inserted neighbors. If there are multiple such places, then we insert the vertex to a place where the objective value increases the least. They insert the vertices in any order in which the vertices of degree at most 2 are last. Intuitively, the same algorithm could work for the problem of maximizing $\sum_{v\in V}\overleftarrow{d}(v)\overrightarrow{d}(v)$, but unfortunately one can construct a sequence of graphs for which the approximation ratio goes to infinity. For example consider the graph sequence shown in Figure 12. For $G_k=(V,E)$, the algorithm outputs the order $u,w_1,\dots,w_{2k},v$, for which $\sum_{v\in V}\overleftarrow{d}(v)\overrightarrow{d}(v)=2k$. However, one can argue that the order $w_1,\dots,w_k,u,v,w_{k+1},\dots,w_{2k}$ with $\sum_{v\in V}\overleftarrow{d}(v)\overrightarrow{d}(v)=2k^2$ is optimal. So the approximation ratio is $\frac{2k^2}{2k}$, which goes to infinity if $k\to\infty$.

In what follows, we show that a random order of the vertices gives an expected 3-approximation for maximizing $\sum_{v\in V}\overleftarrow{d}(v)\overrightarrow{d}(v)$, even in case of multigraphs. After that, we present a deterministic algorithm achieving the same approximation ratio for simple graphs.

5.3 Approximation ratio of orienting by a random permutation

This section investigates the expected approximation ratio of a random order of the vertices for maximizing $\sum_{v\in V}\overleftarrow{d}(v)\overrightarrow{d}(v)$ in case of multigraphs.

Theorem 41 *Let us be given a multigraph $G = (V, E)$. A random permutation of the vertices is a 3-approximate solution in expectation for maximizing the value of $\sum_{v\in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$.*

Proof: Let n denote the number of vertices and $\mathcal{S}_V$ denote the set of all permutations of V. Let $\mathbb{E}(v)$ denote the expected value of $\overleftarrow{d}(v)\overrightarrow{d}(v)$ in a random order, that is, $\mathbb{E}(v) = \frac{\sum_{\sigma\in\mathcal{S}_V} \overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)}{n!}$. By definition, the expected objective value of a random permutation is

$$\frac{\sum_{\sigma\in\mathcal{S}_V}\sum_{v\in V} \overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)}{n!} = \sum_{v\in V}\frac{\sum_{\sigma\in\mathcal{S}_V} \overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)}{n!} = \sum_{v\in V}\mathbb{E}(v).$$

Therefore, it suffices to prove that $\mathbb{E}(v)$ is at least the third of the product $\overleftarrow{d}(v)\overrightarrow{d}(v)$ in an optimal order for each vertex $v \in V$. The statement is clearly true if $d(v) = 0$. Consider a vertex v with $d(v) \geq 1$, and let $\{v_1, \dots, v_d\}$ denote its neighbors. Let x_i denote the number of parallel edges between v and v_i for each $i \in \{1, \dots, d\}$, and assume that $x_1 \leq \cdots \leq x_d$. Let s denote the degree of v with multiplicities, that is, $s = \sum_{i=1}^d x_i$.

We prove that

$$\mathcal{U}(v) = \begin{cases} x_d(s - x_d) & \text{if } x_d > s - x_d, \\ \frac{s^2}{4} & \text{if } x_d \leq s - x_d \end{cases}$$

is an upper bound on $\overleftarrow{d}(v)\overrightarrow{d}(v)$ in the optimal order. To see this, observe that $\overleftarrow{d}(v)\overrightarrow{d}(v) = \overleftarrow{d}(v)(s - \overleftarrow{d}(v))$. This expression is maximized when $\overleftarrow{d}(v)$ is as close to $\frac{s}{2}$ as possible, therefore, $\frac{s^2}{4}$ is an upper bound for $\overleftarrow{d}(v)\overrightarrow{d}(v)$. In the case when $x_d > s - x_d$ (i.e. $x_d > \frac{s}{2}$), if the vertex v_d is on the left side of v then $\overleftarrow{d}(v) \geq x_d$; otherwise, if the vertex v_d is on the right side of v, then $\overleftarrow{d}(v) \leq s - x_d$. So $\overleftarrow{d}(v)$ is the nearest to $\frac{s}{2}$ if $\overleftarrow{d}(v) = x_d$ or $\overleftarrow{d}(v) = s - x_d$, therefore, $\mathcal{U}(v) = x_d(s - x_d)$ is an upper bound.

What we prove is that $\mathbb{E}(v)$ is at least $\frac{1}{3}\mathcal{U}(v)$. The proof is divided into two steps.

Step 1: We prove the statement in the special case when $x_1 = \cdots = x_{d-1} = 1$ and $x_d = k \geq 1$. Observe that for any distinct $i, j \in \{1, \dots, d+1\}$, there are exactly $\frac{|\mathcal{S}_V|}{d(d+1)}$ permutations of all vertices in which v is the i^{th} and v_d is the j^{th} vertex among the vertices $v, v_1, \dots, v_d$. Moreover, in any such permutation,

$$\overleftarrow{d}(v)\overrightarrow{d}(v) = \begin{cases} (i-1)(d-i+k) & \text{if } i < j, \\ (i-2+k)(d-i+1) & \text{if } j < i \end{cases}$$

holds by definition.

Therefore, we can compute the expected value of $\overleftarrow{d}(v)\overrightarrow{d}(v)$ by taking the average of the products $\overleftarrow{d}(v)\overrightarrow{d}(v)$ over all possible placements of v and v_d. Formally,

$$\begin{aligned}
\mathbb{E}(v) &= \frac{\sum_{i=1}^d\sum_{j=i+1}^{d+1}(i-1)(d-i+k) + \sum_{i=2}^{d+1}\sum_{j=1}^{i-1}(i-2+k)(d-i+1)}{d(d+1)} \\
&= \frac{\sum_{i=2}^d (d-i+1)(i-1)(d-i+k) + \sum_{i=2}^d (i-1)(i-2+k)(d-i+1)}{d(d+1)} \\
&= \frac{\sum_{i=2}^d (i-1)(d-i+1)(d+2k-2)}{d(d+1)} = \frac{(d+2k-2)\sum_{i=1}^{d-1} i(d-i)}{d(d+1)} \\
&= (d+2k-2)\left(\frac{\sum_{i=1}^{d-1} i}{d+1} - \frac{\sum_{i=1}^{d-1} i^2}{d(d+1)}\right) = (d+2k-2)\left(\frac{d(d-1)}{2(d+1)} - \frac{(d-1)(2d-1)}{6(d+1)}\right) \\
&= (d+2k-2)(d-1)\left(\frac{3d-(2d-1)}{6(d+1)}\right) = \frac{(d-1)(d+2k-2)}{6},
\end{aligned}$$

where the first equation holds by the argument above, and all other equations follow by rearrangements of the expressions. We proceed to show that $\mathbb{E}(v) \geq \frac{1}{3}\mathcal{U}(v)$.

If $x_d > (s - x_d)$, that is, $k > d - 1$, then

$$\mathbb{E}(v) = \frac{(d-1)(d+2k-2)}{6} \geq \frac{2k(d-1)}{6} = \frac{1}{3}\mathcal{U}(v),$$

where the inequality is true because $d \geq 1$ and $k > d - 1$.

Otherwise, if $x_d \leq (s - x_d)$, that is, $k \leq d - 1$, then

$$\mathbb{E}(v) = \frac{(d-1)(d+2k-2)}{6} \geq \frac{(d+k-1)(d+2k-2)}{12} \geq \frac{(d+k-1)(d+k-1)}{12} = \frac{1}{3}\mathcal{U}(v),$$

where the inequality is true because $d + 1 \geq k$ and $k \geq 1$. This completes Step 1.

Step 2: We prove that $\mathbb{E}(v) \geq \frac{1}{3}\mathcal{U}(v)$ holds for arbitrary $x_1 \leq \cdots \leq x_d$. First, we show a simple modification of the values $x_1, \ldots, x_d$ preserving the inequality to be proven.

Take an index $j \in \{1, \ldots, d-1\}$ for which $x_d \geq x_j + 2$, and consider the sequence $x'_1, \ldots, x'_d$ which is obtained by the following modification of $x_1, \ldots, x_d$:

$$x'_i = \begin{cases} x_i - 1 & \text{if } i = d, \\ x_i + 1 & \text{if } i = j, \\ x_i & \text{otherwise.} \end{cases}$$

We re-index the sequence $x'_1, \ldots, x'_d$ such that $x'_1 \leq \cdots \leq x'_d$ holds. We denote the expected value, the upper bound, the left degree and the right degree of v for the modified sequence $x'_1, \ldots, x'_d$ by $\mathbb{E}'(v)$, $\mathcal{U}'(v)$, $\overleftarrow{d}'(v)$ and $\overrightarrow{d}'(v)$, respectively, while we keep the original notations for the sequence $x_1, \ldots, x_d$.

We prove that $\mathbb{E}(v) \geq \frac{1}{3}\mathcal{U}(v)$ implies $\mathbb{E}'(v) \geq \frac{1}{3}\mathcal{U}'(v)$, by showing that

$$\mathbb{E}'(v) - \mathbb{E}(v) \geq \frac{1}{3}(\mathcal{U}'(v) - \mathcal{U}(v)).$$

For a given order σ, let $\Delta_\sigma(v) = \overleftarrow{d}'_\sigma(v)\overrightarrow{d}'_\sigma(v) - \overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)$. Observe that $\mathbb{E}'(v) - \mathbb{E}(v) = \frac{\sum_{\sigma \in \mathcal{S}_V} \Delta_\sigma(v)}{|\mathcal{S}_V|}$.

On the one hand, if $x_d \leq s - x_d$, then $x'_d = x_d - 1 \leq s - x_d - 1 < s' - x'_d$ also holds, so $\mathcal{U}'(v) = \mathcal{U}(v) = \frac{s^2}{4}$. Therefore, it is enough to show that $\mathbb{E}'(v) - \mathbb{E}(v) = \frac{\sum_{\sigma \in \mathcal{S}_V} \Delta_\sigma(v)}{|\mathcal{S}_V|} \geq 0$ holds. If v_j and v_d are in the same side of v in σ, then $\overleftarrow{d}'_\sigma(v)\overrightarrow{d}'_\sigma(v) = \overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)$, therefore, $\Delta_\sigma(v) = 0$. Consider those orders σ, in which v_d and v_j are in opposite direction from v in σ. Arrange such orders in pairs: let σ and σ' form a pair if we can get σ' by swapping the positions of v_j and v_d in σ. Consider a pair σ, σ' and assume that v_j is on the left side of v in σ and on the right side of v in σ'. Let L denote the sum of the x_i values on the left side of v in σ, and let R denote the sum of the x_i values on the right side of v in σ. By definition, $\Delta_\sigma(v) = (L+1)(R-1) - LR = R - L - 1$ and $\Delta_{\sigma'}(v) = (L - x_j + x_d - 1)(R - x_d + x_j + 1) - (L - x_j + x_d)(R - x_d + x_j) = L - R - 2x_j + 2x_d - 1$, therefore

$$\Delta_\sigma(v) + \Delta_{\sigma'}(v) = R - L - 1 + L - R - 2x_j + 2x_d - 1 = 2x_d - 2x_j - 2 > 0,$$

where the inequality holds because $x_d \geq x_j + 2$. This implies that $\mathbb{E}'(v) - \mathbb{E}(v) = \frac{\sum_{\sigma \in \mathcal{S}_V} \Delta_\sigma(v)}{|\mathcal{S}_V|} > 0$.

On the other hand, if $x_d > s - x_d$, then $\mathcal{U}'(v) - \mathcal{U}(v) = (x_d - 1)(s - x_d + 1) - x_d(s - x_d) = 2x_d - s - 1$. Observe that x_d and x_j are in the same direction from v in two-third of the permutations, which means that $\overleftarrow{d}'_\sigma(v)\overrightarrow{d}'_\sigma(v) - \overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v) = 0$, that is, $\Delta_\sigma(v) = 0$. In the other one-third of the permutations, x_d and x_j are in the opposite direction from v. Consider an order σ in which v_j and v_d are in the opposite direction from v. Denote the sum of the values in the side of v in σ that contains v_d by D. Then the following holds:

$$\Delta_\sigma(v) = (D-1)(s-D+1) - D(s-D) = 2D - s - 1 \geq 2x_d - s - 1.$$

Therefore, in two-third of the permutations $\Delta_\sigma(v) = 0$ holds, and in one-third of the permutations $\Delta_\sigma(v) \geq (2x_d - s - 1)$ holds, which implies that $\mathbb{E}'(v) - \mathbb{E}(v) \geq \frac{1}{3}(2x_d - s - 1) = \frac{1}{3}(\mathcal{U}'(v) - \mathcal{U}(v))$. We conclude that $\mathbb{E}'(v) \geq \frac{1}{3}\mathcal{U}'(v)$.

Observe that any sequence $x_1, \dots, x_d$ is reachable from the sequence $y_1, \dots, y_d$, where $y_1 = \dots = y_{d-1} = 1$ and $y_d = \sum_{i=1}^{d} x_i - (d-1)$, by repeatedly applying the modification described above. The inequality holds for $y_1, \dots, y_d$ by Step 1, and the modification given in Step 2 preserves the inequality, hence it also holds for any $x_1, \dots, x_d$. This completes Step 2, and also the proof of the theorem. $\square$

5.4 De-randomized approximation algorithm

In this section, we give a deterministic polynomial-time 3-approximation algorithm for maximizing $\sum_{v \in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$. Let us be given a multigraph graph $G = (V, E)$, where $|V| = n$. Let $\mathbb{E}_{v_1,\dots,v_i}$ denote the expected objective value under all permutations of the vertex set V in which the first $i \leq n$ vertices are the vertices $v_1, \dots, v_i$ in this order. Formally,

$$\mathbb{E}_{v_1,\dots,v_i} = \frac{\sum_{\sigma \in \mathcal{S}_V, \sigma_j = v_j \forall j \leq i} \sum_{v \in V} \overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)}{(n-i)!},$$

where $v_1, \dots, v_i$ are distinct vertices in V.

Now we are in a position to describe the deterministic algorithm, which can be seen as the de-randomized version of the algorithm given in the previous section using conditional probabilities. The algorithm fixes the vertices from left to right. In the i^{th} step, there are already some vertices $v_1, \dots, v_{i-1}$ fixed at the first $(i-1)$ places. We compute the expected value $\mathbb{E}_{v_1,\dots,v_{i-1},v}$ for each remaining vertex $v \in V \setminus \{v_1, \dots, v_{i-1}\}$, and fix a vertex v at the i^{th} place maximizing the expected value.

First, we prove that we can compute the expected value $\mathbb{E}_{v_1,\dots,v_i}$ for any distinct vertices $v_1, \dots, v_i \in V$ in polynomial time, which immediately implies that the algorithm runs in polynomial time.

Claim 42 *For any distinct $v_1, \dots, v_i \in V$, $\mathbb{E}_{v_1,\dots,v_i}$ can be computed in polynomial time.*

Proof: We call a permutation *relevant* if it begins with the vertices $v_1, \dots, v_i$ in this order. For $v \in V$, let $\mathbb{E}_{v_1,\dots,v_i}(v)$ denote the expected value of $\overleftarrow{d}(v)\overrightarrow{d}(v)$ in a random relevant permutation of V. By definition and simple rearrangements,

$$\mathbb{E}_{v_1,\dots,v_i} = \frac{\sum_{\sigma \in \mathcal{S}_V, \sigma_j = v_j \forall j \leq i} \sum_{v \in V} \overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)}{(n-i)!} = \sum_{v \in V} \frac{\sum_{\sigma \in \mathcal{S}_V, \sigma_j = v_j \forall j \leq i} \overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)}{(n-i)!} = \sum_{v \in V} \mathbb{E}_{v_1,\dots,v_i}(v).$$

Therefore, it is enough to show that we can compute $\mathbb{E}_{v_1,\dots,v_i}(v)$ in polynomial time. Let $V_j = \{v_1, \dots, v_j\}$ for $j \in \{1, \dots, i\}$. First, suppose that $v = v_j$ for some $j \leq i$. This means that $\mathbb{E}_{v_1,\dots,v_i}(v_j) = d(v_j, V_{j-1})d(v_j, V \setminus V_j)$, because the left and right degrees of the vertex v_j remains the same in all relevant permutations.

Second, suppose that $v \in V \setminus V_i$. We give a dynamic programming method for computing $\mathbb{E}_{v_1,\dots,v_i}(v)$ using the fact that only the vertices of $V \setminus V_i$ can appear on the right side of v in any relevant permutation. First we need some notations: Let us denote the neighbors of v from $V \setminus V_i$ by $n_1, \dots, n_d$. Let $f(j, k, \ell)$ denote the number of those permutations of $\{v, n_1, \dots, n_j\}$ in which v has k succeeding vertices and right degree equal to ℓ for $j, k \in \{0, \dots, d\}$ and $\ell \in \{0, \dots, d(v, V \setminus V_i)\}$. Otherwise, let $f \equiv 0$. Observe that $f(0, k, \ell) = \begin{cases} 1 & \text{if } k = \ell = 0, \\ 0 & \text{otherwise.} \end{cases}$

Claim 43 *The following recursion holds for the function $f(j, k, \ell)$:*

$$f(j, k, \ell) = k \cdot f(j-1, k-1, \ell - d(v, \{n_j\})) + (j-k) \cdot f(j-1, k, \ell),$$

Proof: Consider a permutation σ counted in $f(j, k, \ell)$ and note that there exists a unique permutation σ' from which we can get σ by inserting n_j to the right position. Observe that σ' is counted either in $f(j-1, k-1, \ell - d(v, \{n_j\}))$ or in $f(j-1, k, \ell)$, depending on the relative order of n_j and v. We say that

σ is corresponding to σ'. Now observe a permutation σ' counted in $f(j-1,k-1,\ell-d(v,\{n_j\}))$ and note that σ' has exactly k corresponding permutations counted in $f(j,k,\ell)$, because we can insert n_j to one of the k possible positions after v. Similarly, a permutation σ' counted in $f(j-1,k,\ell)$ has exactly $(j-k)$ corresponding permutations counted in $f(j,k,\ell)$, because we can insert n_j to one of the $(j-k)$ possible positions before v. This proves the correctness of the recursion. $\square$

Observe that if we compute the $f(j,k,\ell)$ values in increasing order by j and memoize the results, then the expressions on the right side of the recursion are already accessible in every step. Therefore, we can compute $f(j,k,\ell)$ for all $j,k \in \{0,\dots,d\}$ and $\ell \in \{0,\dots,d(v,V\setminus V_i)\}$ in $O(|V|^2|E|)$ time. Note that $f(k,j,\ell), \le |V|!$, so it can be stored in $O(|V|\log(|V|))$ bits, these together imply that the given dynamic program runs in polynomial time.

Now we show a formula to determine $\mathbb{E}_{v_1,\dots,v_i}(v)$ using the computed $f(d,k,\ell)$ values, where d denotes the number of neighbors of v in $V\setminus V_i$.

By definition, any relevant permutation contains $v_1,\dots,v_i$ on the first i places, and after them the other vertices according to a permutation of $V\setminus V_i$. Therefore, there are $|\mathcal{S}_{V\setminus V_i}|$ relevant permutations and v has right degree equal to ℓ in exactly $\sum_{k=0}^{d} f(d,k,\ell)$ relevant permutations. This means that $\mathbb{E}_{v_1,\dots,v_i}(v)$ is equal to the weighted average of the $(d(v)-\ell)\ell$ values with weights $\sum_{k=0}^{d} f(d,k,\ell)$:

$$\mathbb{E}_{v_1,\dots,v_i}(v) = \frac{\sum_{k=0}^{d} f(d,k,\ell)(d(v)-\ell)\ell}{|\mathcal{S}_{V\setminus V_i}|}$$

Since we can compute this in polynomial time, the de-randomized algorithm also runs in polynomial time. $\square$

Remark 44 *In case of simple graphs v is succeeded by ℓ neighbors in exactly $\frac{|S_{V\setminus V_i}|}{d(v,V\setminus V_i)+1}$ of the relevant permutations, therefore*

$$\mathbb{E}_{v_1,\dots,v_i}(v) = \frac{\sum_{\ell=0}^{d(v,V\setminus V_i)}(d(v)-\ell)\ell}{d(v,V\setminus V_i)+1} = \frac{d(v)\sum_{\ell=0}^{d(v,V\setminus V_i)}\ell}{d(v,V\setminus V_i)+1} - \frac{\sum_{\ell=0}^{d(v,V\setminus V_i)}\ell^2}{d(v,V\setminus V_i)+1}$$
$$= \frac{d(v)d(v,V\setminus V_i)}{2} - \frac{d(v,V\setminus V_i)(2d(v,V\setminus V_i)+1)}{6} = \frac{d(v,V\setminus V_i)(3d(v)-2d(v,V\setminus V_i)-1)}{6}.$$

Hence, in case of simple graphs, we can compute $\mathbb{E}_{v_1,\dots,v_i}(v)$ by evaluating this explicit formula instead of the previous dynamic programming method.

In the rest of this section, we prove that our algorithm finds a 3-approximate order.

Theorem 45 *The de-randomized algorithm is a 3-approximation algorithm for* $\max\sum_{v\in V}\overleftarrow{d}(v)\overrightarrow{d}(v)$.

Proof: Denote the optimum value by opt. Consider the step of the algorithm when the first i vertices, denoted by $v_1,\dots,v_i$, are fixed and denote the set of these vertices by V_i. We prove by induction on i that $\mathbb{E}_{v_1,\dots,v_i} \ge \frac{1}{3}\text{opt}$ holds for each $i \in \{1,\dots,n\}$, that is, fixing the vertices of $V\setminus V_i$ in a random order after $v_1,\dots,v_i$ gives a 3-approximate order in expectation. This proves the theorem for $i=n$.

Theorem 41 implies the claim for $i=0$. Suppose it holds for some $i<n$, so $\mathbb{E}_{v_1,\dots,v_i} \ge \frac{1}{3}\text{opt}$. From the definitions and simple rearrangements, we get the following:

$$\mathbb{E}_{v_1,\dots,v_i} = \frac{\sum_{\sigma\in\mathcal{S}_V,\sigma_j=v_j\forall j\le i}\sum_{v\in V}\overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)}{(n-i)!} = \frac{\sum_{u\in V\setminus V_i}\sum_{\sigma\in\mathcal{S}_V,\sigma_j=v_j\forall j\le i,\sigma_{i+1}=u}\sum_{v\in V}\overleftarrow{d}_\sigma(v)\overrightarrow{d}_\sigma(v)}{(n-i)!}$$
$$= \frac{\sum_{u\in V\setminus V_i}(n-i-1)!\,\mathbb{E}_{v_1,\dots,v_i,u}}{(n-i)!} = \frac{\sum_{u\in V\setminus V_i}\mathbb{E}_{v_1,\dots,v_i,u}}{(n-i)} \le \frac{\sum_{u\in V\setminus V_i}\mathbb{E}_{v_1,\dots,v_{i+1}}}{(n-i)} = \mathbb{E}_{v_1,\dots,v_{i+1}},$$

where all the equations come from the definitions of $\mathbb{E}_{v_1,\dots,v_i}$ and $\mathbb{E}_{v_1,\dots,v_i,v}$ and rearrangements of the expressions, and the inequality is true, because the algorithm chooses a vertex v_{i+1} for which the expected objective value $\mathbb{E}_{v_1,\dots,v_i,u}$ is maximal. So $\mathbb{E}_{v_1,\dots,v_i,v_{i+1}} \ge \mathbb{E}_{v_1,\dots,v_i} \ge \frac{1}{3}\text{opt}$ holds. $\square$

6 Open questions

In Section 3, we gave a comprehensive analysis of the complexities of different lexicographically optimal acyclic orientation problems by considering the vertex-ordering alternatives of the problems. We proved that the same orders are optimal for the dec-min 2-bounded and inc-max 2-bounded ordering problems, but the complexity of finding such an optimal order remains open. We proved the NP-hardness of the dec-min and inc-max k-bounded ordering problems for any $k \geq 3$. This implies the NP-hardness of the dec-min problem for k-degenerate graphs for any $k \geq 3$, but we do not know the complexity of the inc-max problem in case of k-degenerate graphs. As one of our main result, we proved in Section 4 that the min $\sum_{v \in V} h(\varrho(v))$ acyclic orientation problem is NP-hard for any discrete strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$ if parallel edges are allowed in the graph. However, the complexity remains open in case of simple graphs. For the special case of $h(z) = z^2$, we extended the NP-hardness proof for simple graphs. We examined the approximation ratio of a greedy algorithm for this special case. The worst approximation ratio of graph sequence that we found tends to $\frac{9}{7}$. It is an open question whether the greedy algorithm approximates the problem within a constant factor. The routing application described in Section 1.1 uses the dec-min and min $\sum_{v \in V} h(\overleftarrow{d}(v))$ acyclic orientation problems with the modification that we are searching for a rooted-connected acyclic orientation. However, the complexities of the egalitarian acyclic orientation problems remain open in this constrained version. In Section 5, we showed that orienting the edges from left to right in a random vertex order gives a 3-approximate solution in expectation for the max $\sum_{v \in V} \varrho(v)\delta(v)$ acyclic orientation problem even for multigraphs. It remains open whether a better approximation exists.

References

[1] P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1-2):123–134, 2000.

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

[3] Y. Asahiro, E. Miyano, H. Ono, and K. Zenmyo. Graph orientation algorithms to minimize the maximum outdegree. *International Journal of Foundations of Computer Science*, 18(02):197–215, 2007.

[4] P. R. Berman, A. D. Scott, and M. Karpinski. Approximation hardness and satisfiability of bounded occurrence instances of SAT. Technical report, SIS-2003-269, 2003.

[5] T. Biedl, T. Chan, Y. Ganjali, M. T. Hajiaghayi, and D. R. Wood. Balanced vertex-orderings of graphs. *Discrete Applied Mathematics*, 148(1):27–48, 2005.

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

[7] M. Burcea, W. K. Hon, H. H. Liu, P. W. K. Wong, and D. K. Y. Yau. Scheduling for electricity cost in a smart grid. *Journal of Scheduling*, 19:687–699, 2016.

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

[9] H. de Fraysseix and P. O. de Mendez. Regular orientations, arboricity, and augmentation. In *Graph Drawing*, pages 111–118. Springer, 1995.

[10] S. Even and R. E. Tarjan. Computing an *st*-numbering. *Theoretical Computer Science*, 2(3):339–344, 1976.

[11] A. Frank. On the orientation of graphs. *Journal of Combinatorial Theory, Series B*, 28(3):251–261, 1980.

[12] A. Frank and K. Murota. Decreasing minimization on M-convex sets: algorithms and applications. *Mathematical Programming*, 195(1-2):1027–1068, 2022.

[13] A. Frank and K. Murota. Decreasing minimization on M-convex sets: background and structures. *Mathematical Programming*, 195(1-2):977–1025, 2022.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Company, 1979.

[15] S. L. Hakimi. On the degrees of the vertices of a directed graph. *Journal of the Franklin Institute*, 279(4):290–308, 1965.

[16] J. Kára, J. Kratochvíl, and D. R. Wood. On the complexity of the balanced vertex ordering problem. In *International Computing and Combinatorics Conference*, pages 849–858. Springer, 2005.

[17] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[18] Z. Király and D. Pálvölgyi. Acyclic orientations with degree constraints. arXiv:1806.03426, 2018.

[19] J. Kratochvıl and Z. Tuza. On the complexity of bicoloring clique hypergraphs of graphs. *Journal of Algorithms*, 45(1):40–54, 2002.

[20] D. R. Lick and A. T. White. k-degenerate graphs. *Canadian Journal of Mathematics*, 22(5):1082–1096, 1970.

[21] D. W. Matula and L. L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM (JACM)*, 30(3):417–427, 1983.

[22] R. R. Meyer. A class of nonlinear integer programs solvable by a single linear program. *SIAM Journal on Control and Optimization*, 15(6):935–946, 1977.

[23] J. C. Sancho, A. Robles, and J. Duato. A new methodology to compute deadlock-free routing tables for irregular networks. In *International Workshop on Communication, Architecture, and Applications for Network-Based Parallel Computing*, pages 45–60. Springer, 2000.

[24] M. D. Schroeder, A. D. Birrell, M. Burrows, H. Murray, R. M. Needham, T. L. Rodeheffer, E. H. Satterthwaite, and C. P. Thacker. Autonet: A high-speed, self-configuring local area network using point-to-point links. *IEEE Journal on Selected Areas in Communications*, 9(8):1318–1335, 1991.

[25] É. Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5(3):247–255, 1985.

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

[27] V. W. Wittorff. Implementation of constraints to ensure deadlock avoidance in networks, May 12 2009. US Patent 7,532,584.

[28] H. Zhou and X. Hou. Strongly connected orientation with minimum lexicographic order of indegrees. *International Journal of Foundations of Computer Science*, 33(02):149–153, 2022.