SIAM J. DISCRETE MATH.
Vol. 32, No. 3, pp. 2067–2079

OPTIMIZATION OVER DEGREE SEQUENCES*

ANTOINE DEZA†, ASAF LEVIN‡, SYED M. MEESUM§, AND SHMUEL ONN‡

Abstract. We introduce and study the problem of optimizing arbitrary functions over degree sequences of hypergraphs and multihypergraphs. We show that over multihypergraphs the problem can be solved in polynomial time. For hypergraphs, we show that deciding whether a given sequence is the degree sequence of a 3-hypergraph is NP-complete, thereby solving a 30 year long open problem. This implies that optimization over hypergraphs is hard even for simple concave functions. In contrast, we show that for graphs, if the functions at vertices are the same, then the problem is polynomial time solvable. We also provide positive results for convex optimization over multihypergraphs and graphs and exploit connections to degree sequence polytopes and threshold graphs. We then elaborate on connections to the emerging theory of shifted combinatorial optimization.

Key words. graph, hypergraph, combinatorial optimization, degree sequence, threshold graph, extremal combinatorics, computational complexity

AMS subject classifications. 05A, 15A, 51M, 52A, 52B, 52C, 62H, 68Q, 68R, 68U, 68W, 90B, 90C

DOI. 10.1137/17M1134482

1. Introduction.

1. Introduction. The *degree sequence* of a (simple) graph $G = (V, E)$ with $V = [n] := \{1, \ldots, n\}$ and $m = |E|$ edges is the vector $d = (d_1, \ldots, d_n)$ with $d_i = |\{e \in E : i \in e\}|$ the degree of vertex i for all i.

Degree sequences have been studied by many authors, including the celebrated work of Erdős and Gallai [4] from 1960, which effectively characterizes the degree sequences of graphs. See, for example, [5] and the references therein for some of the more recent work in this area.

In this article, we are interested in the following discrete optimization problem. Given n, m, and functions $f_i : \{0, 1, \ldots, m\} \to \mathbb{R}$ for $i = 1, \ldots, n$, find a graph on $[n]$ with m edges having degree sequence d maximizing $\sum_{i=1}^{n} f_i(d_i)$. The special case with $f_i(t) = t^2$ for all i, that is, finding a graph maximizing the sum of degree squares, was solved previously in [20].

More generally, we are interested in the problem over (uniform) hypergraphs. In what follows, it will be convenient to use a vector notation, and so we make the following definitions. A *k-hypergraph with m edges on $[n]$* is a subset $H \subseteq \{0,1\}^n_k := \{x \in \{0,1\}^n : \|x\|_1 = k\}$ with $|H| = m$ (keeping in mind also the interpretation of H as the set of supports of its vectors). We also consider the problem over multihypergraphs. A *k-multihypergraph with m edges on $[n]$* is a matrix $H \in (\{0,1\}^n_k)^m$, that is, an $n \times m$ matrix with each column $H^j \in \{0,1\}^n_k$ representing an edge (that is, each column is

*Received by the editors June 13, 2017; accepted for publication (in revised form) June 7, 2018; published electronically August 7, 2018.
http://www.siam.org/journals/sidma/32-3/M113448.html
Funding: The first author's research was partially supported by the Natural Sciences and Engineering Research Council of Canada Discovery grant programs (RGPIN-2015-06163). The second author's research was partially supported by a grant from GIF, the German-Israeli Foundation for Scientific Research and Development (grant I-1366-407.6/2016). The third author's research was partially supported by a grant at the Technion. The fourth author's research was partially supported by the Dresner Chair at the Technion.
†McMaster University, Hamilton, ON L8S 4K1, Canada (deza@mcmaster.ca).
‡Technion - Israel Institute of Technology, Haifa, 3200003 Israel (levinas@ie.technion.ac.il, onn@ie.technion.ac.il).
§Institute of Mathematical Sciences, HBNI, Chennai 600113, India (meesum@imsc.res.in).

a 0-1 vector containing exactly k ones), so that multiple (identical) edges are allowed (but no loops). The *degree sequence of* H is the vector $d = \sum H := \sum\{x : x \in H\}$ for hypergraphs and $d = \sum H := \sum_{j=1}^{m} H^j$ for multihypergraphs.

We pose the following algorithmic problem.

Optimization over degree sequences. Given k, n, m and univariate functions $f_i : \{0, \dots, m\} \to \mathbb{R}$, find a k-(multi)hypergraph H with m edges whose degree sequence $d := \sum H$ maximizes $\sum_{i=1}^{n} f_i(d_i)$.

The case of linear functions will be discussed in section 3. For arbitrary functions and multihypergraphs we solve the problem completely in section 4.

THEOREM 1.1. *The general optimization problem over degree sequences of multihypergraphs can be solved in polynomial time for any k, n, m and any univariate functions $f_1, \dots, f_n$.*

For hypergraphs the problem is much harder. On the positive side we show in section 5 the following theorem, broadly extending the result for sum of degree squares of graphs.

THEOREM 1.2. *For $k = 2$, that is, graphs, the optimization problem over degree sequences can be solved in $O(n^5m^2)$ time for any n, m and any identical univariate functions $f_1 = \cdots = f_n$.*

When the identical functions are *convex* we establish better time complexity in Theorem 6.3.

In order to obtain our result on the negative side, we consider also the following decision problem: given k and $d \in \mathbb{Z}_+^n$, is d the degree sequence of some hypergraph $H \subseteq \{0,1\}_k^n$? For $k = 1$ it is trivial, as d is a degree sequence if and only if $d \in \{0,1\}^n$. For $k = 2$ it is solved by the aforementioned theorem of Erdős and Gallai [4], which implies that d is a degree sequence of a graph if and only if $\sum d_i$ is even and, permuting d so that $d_1 \geq \cdots \geq d_n$, the inequalities $\sum_{i=1}^{j} d_i - \sum_{i=l+1}^{n} d_i \leq j(l-1)$ hold for $1 \leq j \leq l \leq n$, yielding a polynomial time algorithm.

For $k = 3$ it was raised 30 years ago by Colbourn, Kocay, and Stinson [3, Problem 3.1] and has remained open to date. We solve it in section 2, implying that it is unlikely that degree sequences of 3-hypergraphs could be effectively characterized, in the following theorem.

THEOREM 1.3. *It is NP-complete to decide whether $d \in \mathbb{Z}_+^n$ is the degree sequence of a* 3*-hypergraph.*

This leads at once to the following negative statement by presenting an optimization problem of simple concave functions over degree sequences of 3-hypergraphs whose optimal value is zero if and only if a given $d \in \mathbb{Z}_+^n$ is a degree sequence of some 3-hypergraph.

COROLLARY 1.4. *For $k = 3$ the optimization problem over degree sequences of* 3*-hypergraphs is NP-hard even for concave functions of the form $f_i(t) = -(t - d_i)^2$ with $d_i \in \mathbb{Z}_+$ for each i.*

Next, in section 6, we discuss optimization of convex functions over degree sequences. In fact, our results hold for the more general problem of maximizing any convex function $f : \{0, 1, \dots, m\}^n \to \mathbb{R}$ which is not necessarily separable, as considered above. For this we discuss the *degree sequence polytopes* studied in [12, 13, 17, 19, 21] and references therein, introduce and study degree sequence polytopes of hypergraphs with prescribed numbers of edges, and show that for $k = 2$ their vertices correspond to suitable *threshold graphs* [18].

Finally, in section 7, we illustrate that optimization over degree sequences can be viewed within the framework of *shifted combinatorial optimization* recently introduced and investigated in the series of papers [6, 11, 14, 16] and contributes to this emerging new theory.

2. The complexity of deciding hypergraph degree sequences. Here we consider the complexity of deciding the existence of a hypergraph with a given degree sequence. We prove the following theorem, solving a problem raised 30 years ago by Colbourn, Kocay, and Stinson [3]. The proof is partially inspired by an argument from [17].

Theorem 1.3. It is NP-complete to decide whether $d \in \mathbb{Z}_+^n$ is the degree sequence of a 3-hypergraph.

Proof. The problem is in NP since if d is a degree sequence, then a hypergraph $H \subseteq \{0,1\}_3^n$ of cardinality $|H| \leq \binom{n}{3} = O(n^3)$ can be exhibited and $d = \sum H$ verified in polynomial time.

Now consider the following so-called 3-*partition* problem: given $a \in \mathbb{Z}_+^n$ and $b \in \mathbb{Z}_+$, decide whether there is an $H \subseteq \{0,1\}_3^n$ such that $ax := \sum_{i=1}^n a_i x_i = b$ for all $x \in H$ and $\sum H = \mathbf{1}$, where $\mathbf{1}$ is the all-ones vector. It is well known to be NP-complete [7], and we reduce it to ours. (In [7], the problem is given in an equivalent form in terms of sets rather than vectors, and even the special case with $n = 3m$ for some m, $a\mathbf{1} = mb$, and $\frac{1}{4}b < a_i < \frac{1}{2}b$ for all i is NP-complete.)

Given such a and b, define $w \in \mathbb{Z}^n$ by $w_i := 3a_i - b$ for all i. Then for any $x \in \{0,1\}_3^n$ we have $wx = 3ax - b\sum_{i=1}^n x_i = 3(ax-b)$, and so $wx = 0$ if and only if $ax = b$. So H satisfies $ax = b$ for all $x \in H$ and $\sum H = \mathbf{1}$ if and only if $wx = 0$ for all $x \in H$ and $\sum H = \mathbf{1}$. For this to hold we must have $w\mathbf{1} = w\sum H = \sum\{wx : x \in H\} = 0$. So if $w\mathbf{1} \neq 0$, then there is no solution to the 3-partition problem and we can define d to be a unit vector in $\mathbb{R}^n$ making sure there is no 3-hypergraph with degree sequence d as well. So we may and do assume $w\mathbf{1} = 0$.

For $\sigma \in \{-,0,+\}$ define a hypergraph $S_\sigma := \{x \in \{0,1\}_3^n : \operatorname{sign}(wx) = \sigma\}$ so that these three hypergraphs form a partition $S_- \uplus S_0 \uplus S_+ = \{0,1\}_3^n$ of the complete 3-hypergraph.

Define $d := \mathbf{1} + \sum S_+$, and observe that w, S_+, and d can be computed in polynomial time. We claim that there is a 3-hypergraph G with degree sequence d if and only if there is a 3-hypergraph H with $wx = 0$ for all $x \in H$ and $\sum H = \mathbf{1}$. So the 3-partition problem reduces to deciding degree sequences, showing that the latter is NP-complete.

We now prove the claim. Suppose first that H satisfies $wx = 0$ for all $x \in H$ and $\sum H = \mathbf{1}$. Then $H \subseteq S_0$, and so $H \cap S_+ = \emptyset$. Let $G := H \uplus S_+$. Then $\sum G = \sum H + \sum S_+ = \mathbf{1} + \sum S_+ = d$, so G has degree sequence d. Conversely, suppose G has degree sequence d. Then, using $w\mathbf{1} = 0$,

$$w\sum G = \sum_{x \in G\cap S_-} wx + \sum_{x \in G\cap S_0} wx + \sum_{x \in G\cap S_+} wx \leq \sum_{x \in S_+} wx = w\mathbf{1} + w\sum S_+ = wd$$

with equality if and only if $G \cap S_- = \emptyset$ and $G \cap S_+ = S_+$ since $\operatorname{sign}(wx) = \sigma$ for each $\sigma \in \{-,0,+\}$ and every $x \in S_\sigma$. Since $\sum G = d$, we do have equality $w\sum G = wd$ above. Let $H := G \cap S_0$. Then $wx = 0$ for all $x \in H$ and $\sum H = \sum G - \sum S_+ = \mathbf{1}$ as claimed. ∎

This result can be easily extended to k-hypergraphs for all fixed $k \geq 3$ using a reduction from the problem for 3-hypergraphs into the problem for k-hypergraphs.

COROLLARY 2.1. *Fix any $k \geq 3$. It is NP-complete to decide whether $d \in \mathbb{Z}_+^n$ is the degree sequence of a k-hypergraph.*

Proof. Given an input $d = (d_1, d_2, \dots, d_n)$ to the problem for 3-hypergraphs, let $m = \frac{1}{3}\sum_{i=1}^n d_i$ and define an input $d' = (d'_1, d'_2, \dots, d'_{n+k-3})$ to the problem for k-hypergraphs where $d'_i = d_i$ for $1 \leq i \leq n$ and $d'_i = m$ for $i > n$. Then any k-hypergraph with degree sequence d' has m edges and each must contain vertices $n+1, \dots, n+k-3$. It can then be verified that d is a degree sequence of a 3-hypergraph if and only if d' is a degree sequence of a k-hypergraph. ∎

3. Linear functions. Here we discuss as a warmup to the following sections the case of linear functions, that is, with $f_i(t) = w_i t$ for each i, where $w = (w_1, \dots, w_n)$ is a given profit vector. So the degree sequence optimization problems are

$$\max\left\{w\sum H \;:\; H \in (\{0,1\}_k^n)^m\right\}, \qquad \max\left\{w\sum H \;:\; H \subseteq \{0,1\}_k^n,\ |H| = m\right\}.$$

PROPOSITION 3.1. *The linear optimization problem over degree sequences of multihypergraphs and over degree sequences of hypergraphs can be solved in polynomial time for all n, k, m.*

Proof. First, consider the case of multihypergraphs. Since $w\sum H = \sum_{j=1}^m wH^j$ and all columns H^j can be the same, an optimal solution will be a matrix $H = [x, \dots, x]$ with all columns equal to some $x \in \{0,1\}_k^n$ maximizing wx. Such an x can be found by sorting w; that is, if for a permutation π of $[n]$ we have $w_{\pi(1)} \geq \cdots \geq w_{\pi(n)}$, then we can take $x := \mathbf{1}_{\pi(1)} + \cdots + \mathbf{1}_{\pi(k)}$, where $\mathbf{1}_i$ denotes the standard ith unit vector in $\mathbb{R}^n$.

Next, consider the case of hypergraphs. Since $w\sum H = \sum\{wx : x \in H\}$, an optimal solution will be a hypergraph H consisting of the m vectors $x \in \{0,1\}_k^n$ with largest values wx. For k fixed we can simply compute wx for each of the $O(n^k)$ vectors in $\{0,1\}_k^n$ in polynomial time and pick the best m vectors. For variable k we can use the algorithm of Lawler [15] to find the m vectors $x \in \{0,1\}_k^n$ with largest values wx in time polynomial in n, k, m. ∎

4. Arbitrary functions over multihypergraphs. The algorithmic results of our work are mostly obtained by the use of dynamic programming. We will use a model of dynamic programming, where there is a finite set of *states* that is partitioned into *stages* (the stages are indexed by $\{1, 2, \dots, n\}$). For each state there is a finite set of *actions* that can be performed. Given a state and an action, the decision maker is moving to a state of a (strictly) larger stage, and incurs a *reward* that is a function of the state and the action. We have an initial state, and a set of final states, and the goal is to find a path maximizing the total reward starting in the initial state and ending at a final state. This optimization problem is equivalent to finding a longest path in a directed acyclic graph, a task known to be solvable with time complexity that is linear in the number of arcs and nodes of that directed graph.

For multihypergraphs there is a characterization of degree sequences and a greedy procedure for constructing a hypergraph from its degree sequence. The characterization and algorithm follow from results of [22] on 0-1 matrices and were also proved in [1, 2]. We record these facts in the following proposition and provide a short proof for completeness.

PROPOSITION 4.1. *Vector $d \in \mathbb{Z}_+^n$ is a degree sequence of k-multihypergraph H with m edges if and only if $\sum_{i=1}^n d_i = km$ and $d_i \leq m$ for all i, and H is polynomial time realizable from d.*

Proof. The conditions on d are clearly necessary. We prove by induction on m that given d satisfying the conditions we can construct a multihypergraph H with $d = \sum H$. If $m = 0$, this is the empty multihypergraph. Suppose $m \geq 1$. Permuting the d_i we may assume $d_1 \geq \cdots \geq d_n$. We claim that $d_k \geq 1$ and $d_{k+1} \leq m - 1$. If $d_k = 0$, then $km = \sum d_i \leq (k-1)m$, which is impossible. If $d_{k+1} \geq m$, then $km = \sum d_i \geq (k+1)m$, which is again impossible. Define a vector d' by $d'_i := d_i - 1$ for $i \leq k$ and $d'_i := d_i$ for $i > k$. This d' satisfies the conditions with $m - 1$, and by induction we can construct a multihypergraph $[H^1, \ldots, H^{m-1}]$ with degree sequence d'. Then $H := [H^1, \ldots, H^{m-1}, H^m]$ with $H^m := \mathbf{1}_1 + \cdots + \mathbf{1}_k$ is the desired multihypergraph with degree sequence d. So the induction follows and hence we are done. ∎

We can now prove our theorem on optimization over degree sequences of multihypergraphs.

Theorem 1.1. The general optimization problem over degree sequences of multihypergraphs can be solved in polynomial time for any k, n, m and any univariate functions $f_1, \ldots, f_n$.

Proof. By Proposition 4.1, we need to solve the following integer programming problem,

$$\max\left\{ \sum_{i=1}^n f_i(d_i) \ : \ d \in \mathbb{Z}_+^n \, , \ \sum_{i=1}^n d_i = km \, , \ d_i \leq m \, , \ i = 1, \ldots, n \right\} ,$$

find an optimal d, and then use the algorithm of Proposition 4.1 to find an H with $d = \sum H$.

An optimal d can be found using dynamic programming. There are n stages, where the decision at stage i is $0 \leq d_i \leq m$ with reward $f_i(d_i)$. The state at the end of stage i is $0 \leq s_i \leq km$ representing the partial sum $s_i = \sum_{j=1}^i d_j$ starting with $s_0 := 0$. Let $f_i^*(s_i)$ be the maximum total reward of a path leading from the initial state s_0 to state s_i in stage i. The recursively computable optimal value at state s_i is given by

$$f_0^*(s_0) \ := 0 \ , \quad f_i^*(s_i) \ := \ \max\{f_{i-1}^*(s_{i-1}) + f_i(d_i) \ : \ s_i = s_{i-1} + d_i\} \ , \quad i = 1, \ldots, n \ .$$

The optimal value is $f_n^*(km)$, and the optimal decisions and the sequence of states on the optimal path can be reconstructed backwards starting with $s_n^* := km$ recursively by

$$(s_{i-1}^*, d_i^*) \ \in \ \arg\max\{f_{i-1}^*(s_{i-1}) + f_i(d_i) \ : \ s_i^* = s_{i-1} + d_i\} \ , \quad i = n, \ldots, 1 \ .$$

Clearly this is doable in time polynomial in k, n, m for any univariate functions $f_1, \ldots, f_n$. This last claim follows since there are $O(n)$ stages and in each stage there are $O(mk)$ states and the computation of f_i^* for a given state in stage i takes $O(m)$ (for all i), and thus the time complexity of this algorithm is $O(nkm^2)$. ∎

5. Identical functions over graphs. In this section, we restrict our attention to graphs. We use alternatively the interpretation of a graph as $G \subseteq \{0,1\}_2^n$ and $G = ([n], E)$. We note the following characterization of degree sequences of graphs from [9, 10] which leads to a greedy procedure for constructing a graph from its degree sequence. A vector $d \in \mathbb{Z}_+^n$ is called *reducible* if, permuting it such that $d_1 \geq \cdots \geq d_n$, we have that $d_{d_1+1} \geq 1$. The *reduction* of d is the vector $d' \in \mathbb{Z}_+^{n-1}$ defined by $d' := (d_2 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n)$. We record the characterization and algorithm in the next proposition and provide a short proof for completeness; see, e.g., [24] for more details.

PROPOSITION 5.1. *Vector $d \in \mathbb{Z}_+^n$ is a degree sequence of graph G if and only if d is reducible with reduction also a degree sequence. So G is polynomial time recursively realizable from d.*

Proof. Assume $d_1 \geq \cdots \geq d_n$. Suppose d is reducible and its reduction d' is the degree sequence of a graph G' on $[n] \setminus \{1\}$. Then the graph G obtained from G' by adding vertex 1 and connecting it to vertices $2, \ldots, d_1 + 1$ has degree sequence d.

Conversely, suppose d is the degree sequence of $G = ([n], E)$. We show that it is also the degree sequence of a graph with 1 as a neighbor of $2, \ldots, d_1 + 1$, which will show that d is reducible and its reduction is also a degree sequence. Call $i \neq j$ a *bad pair* if $\{1, i\} \in E$ and $\{1, j\} \notin E$ but $d_i < d_j$. Then there must be a k with $\{i, k\} \notin E$ and $\{j, k\} \in E$. Then the graph obtained from G by dropping edges $\{1, i\}$, $\{j, k\}$ and adding $\{1, j\}$, $\{i, k\}$ has the same degree sequence but fewer bad pairs. Repeating this procedure we arrive at a graph with degree sequence d and no bad pairs, which implies that it has 1 as a neighbor of $2, \ldots, d_1 + 1$. ◻

We can now prove our theorem on optimization over degree sequences of graphs.

Theorem 1.2. For $k = 2$, that is, graphs, the optimization problem over degree sequences can be solved in $O(n^5 m^2)$ time for any n, m and any identical univariate functions $f_1 = \cdots = f_n$.

Proof. Since the functions f_i are assumed identical, we may optimize over *sorted* degree sequences $d_1 \geq \cdots \geq d_n$, and in fact, this is the reason our approach needs this assumption. For simplicity of notation we denote by f the common function $f_1 = \cdots = f_n = f$. By Proposition 5.1 and by the original version of the Erdős–Gallai characterization of sorted degree sequences, we need to solve the following nonlinear integer optimization problem,

$$\begin{array}{ll} \max & \displaystyle\sum_{i=1}^{n} f(d_i) \\ \text{s.t.} & d \in \mathbb{Z}_+^n\,, \quad d_1 \geq \cdots \geq d_n\,, \quad \sum d_i = 2m\,, \\ & \displaystyle\sum_{i=1}^{j} d_i - j(j-1) \leq \sum_{i=j+1}^{n} \min\{j, d_i\}\,, \quad j = 1, \ldots, n\,, \end{array}$$

find an optimal d, and then use the algorithm of Proposition 5.1 to find a graph G with $d = \sum G$.

Finding an optimal d can be done again by dynamic programming. This time, in every computational path, there are at most n stages, where the decision in stage i is the value of d_i as well as the number α_i of vertices with degree exactly i (of indices $\beta_i + 1, \ldots, \beta_i + \alpha_i$). The reward of this decision is $f(d_i) + \alpha_i \cdot f(i)$. A *final state* in the dynamic programming table is a state corresponding to a value of i that equals $\max\{j : d_j > j\}$ (in this case, i is the largest index of a vertex with degree larger than i, and thus $\beta_i \geq i$, while $i + 1$ has degree at most $i + 1$, and thus by definition $\beta_{i+1} \leq i$, and by the monotonicity of the sequence of β's we conclude that $\beta_i = i$

and, furthermore, that a final state means that the total degrees of all vertices is $2m$). Our goal is to find a computational path (corresponding to a path of decisions) that maximizes the total reward of the decisions along the path starting at the initial state and ending at a final state.

The definition of the states as well as the transition function will enforce both the monotonicity constraints and the Erdős–Gallai constraints. Thus, the states of the dynamic programming table correspond to 5-tuples $(i, p_i, d_i, \beta_i, s_i)$. The meaning of reaching this state is that so far we decided upon the degrees of nodes $1, 2, \ldots, i$, all of them are at least d_i (as we enforce the monotonicity constraints by induction on the length of the path), and their sum is p_i; and we also computed the degrees of nodes $\beta_i + 1$, $\beta_i + 2, \ldots, n$ which are the vertices of degrees at most i, and their sum is s_i.

When we are at state $(i, p_i, d_i, \beta_i, s_i)$ such that $i < \beta_i$ we should decide the value of d_{i+1} and the value of β_{i+1}. As mentioned above, a feasible value of the pair (d_{i+1}, β_{i+1}) has a reward of $f(d_{i+1}) + (\beta_i - \beta_{i+1}) \cdot f(i+1)$. If such a decision is feasible, then it will result in a transition to the state $(i+1, p_i + d_{i+1}, d_{i+1}, \beta_{i+1}, s_i + (i+1) \cdot (\beta_i - \beta_{i+1}))$.

It remains to define the set of pairs (d_{i+1}, β_{i+1}) of nonnegative integers for which the decision is feasible. A pair (d_{i+1}, β_{i+1}) is *feasible* for a given state $(i, p_i, d_i, \beta_i, s_i)$ if all the following conditions hold:

1. $\beta_{i+1} \geq i + 1$: This condition is required as we determine the degrees of both a prefix and a suffix of the vertices, and we need this definition to be well-defined. Using this condition we enforce that the degree of a vertex will be defined at most once either as part of the prefix or as part of the suffix.
2. $i + 1 \leq d_{i+1} \leq d_i$: This constraint enforces the monotonicity conditions over the prefix of the first $i + 1$ vertices (note that d_i is defined in the definition of the state).
3. $p_i + d_{i+1} + s_i + (i+1) \cdot (\beta_i - \beta_{i+1}) \leq m$: This condition will enforce that we can choose at most m edges.
4. The $(i + 1)$th Erdős–Gallai constraint that can be stated as follows:

$$p_i + d_{i+1} - (i+1) \cdot i \leq s_i + (i+1) \cdot (\beta_i - \beta_{i+1}) + (\beta_{i+1} - i - 1) \cdot (i+1).$$

The *initial state* of the dynamic programming is $(0, 0, n, n, 0)$, and we would like to find a path of feasible decisions with maximum total reward leading from the initial state to a final state that is a state for which both $i = \beta_i$ and $p_i + s_i = m$.

Observe that if the decision is feasible and leads to a state that has a path P to a final state, then the decisions that will correspond to the path P will not change the fact that the degree sequence satisfies the $(i + 1)$th Erdős–Gallai constraint. In order to establish this property we examine the change in the two sides of the $(i + 1)$th Erdős–Gallai constraint when we determine the degrees in the next stages. First, consider the left-hand side and note that the left-hand side of this inequality is not changed, as all values that appear in the left-hand side were determined in the previous stages. Consider the right-hand side of this inequality: We argue that it also does not change. This claim holds, as all the degrees that we select in the following stages will be between $i + 2$ and d_{i+1} so that the minimum on the right-hand side for the indices between $i + 2$ and β_{i+1} will be $i + 1$, and this is the value we used in the last condition we checked for the feasibility of the decision.

Next, we note that the number of states of the dynamic programming table is $O(n^3 m^2)$, as there are $O(n)$ options for i, $O(m)$ options for p_i, $O(n)$ options for d_i, $O(n)$ options for β_i, and $O(m)$ options for s_i. Furthermore, for each state there are $O(n^2)$ feasible decisions, and the time for computing the maximum total reward of a subpath that starts at the given state and ends at some final state is $O(n^2)$ (since given a value of β_{i+1} we can compute the upper bound on d_{i+1} resulting from all our constraints in time $O(n)$). Thus, the time complexity of the algorithm is $O(n^5m^2)$. ∎

6. Convex functions and degree sequence polytopes. Here we consider optimization of convex functions over degree sequences. In fact, our results hold for the more general problem of maximizing any convex function $f : \{0, 1, \ldots, m\}^n \to \mathbb{R}$ which is not necessarily separable of the form $f(d_1, \ldots, d_n) = \sum_{i=1}^n f_i(d_i)$ considered before.

First, we consider the case of multihypergraphs.

THEOREM 6.1. *If f is convex, then there exists a multihypergraph $H = [x, \ldots, x]$, having m identical edges x, which maximizes $f(\sum H)$, where x maximizes $f(mx)$ over $\{0,1\}^n_k$. So for any fixed k the optimization problem over degree sequences can be solved in polynomial time. On the other hand, for k variable the problem may need exponential time even for $m = 1$.*

Proof. Assume f is convex, and let $\hat{H}$ be an optimal multihypergraph. Let $x^1, \ldots, x^r$ in $\{0,1\}^n_k$ be the distinct edges of $\hat{H}$. Define an $n \times r$ matrix $M := [x^1, \ldots, x^r]$ and a function $g : \mathbb{Z}^r \to \mathbb{R}$ by $g(y) := f(My)$, which is convex since f is. Consider the integer simplex $Y := \{y \in \mathbb{Z}^r_+ : y_1 + \cdots + y_r = m\}$ and the auxiliary problem $\max\{g(y) : y \in Y\}$. For each $y \in Y$ let $H(y) := [x^1, \ldots, x^1, \ldots, x^r, \ldots, x^r]$ be the multihypergraph consisting of y_i copies of x^i. Permuting the columns of $\hat{H}$ we may assume that $\hat{H} = H(\hat{y})$ for some $\hat{y} \in Y$.

Now g is convex, so the auxiliary problem has an optimal solution which is a vertex of Y, namely, a multiple $\tilde{y} = m\mathbf{1}_i$ of a unit vector in $\mathbb{Z}^r$. It then follows that $\tilde{H} := H(\tilde{y}) = [x^i, \ldots, x^i]$ is the desired optimal multihypergraph having a degree sequence which maximizes f since

$$f\left(\sum_{j=1}^{m} \tilde{H}^j\right) \;=\; f(M\tilde{y}) \;=\; g(\tilde{y}) \;\geq\; g(\hat{y}) \;=\; f(M\hat{y}) \;=\; f\left(\sum_{j=1}^{m} \hat{H}^j\right) \;.$$

For k fixed we can clearly find an x attaining $\max\{f(mx) : x \in \{0,1\}^n_k\}$ in polynomial time.

Now consider the situation of the k variable part of the input. Let $m := 1$. Then any function $f : \{0,1\}^n \to \mathbb{R}$ is convex, so in order to solve $\max\{f(x) : x \in \{0,1\}^n_k\}$ one needs to check the value of f on each of the $\binom{n}{k}$ points in $\{0,1\}^n_k$, and for $k = \lfloor \frac{n}{2} \rfloor$ this requires exponential time. ∎

We continue with hypergraphs. We need to discuss the class of *degree sequence polytopes*. The classical degree sequence polytope is the convex hull $D^n_k := \mathrm{conv}\{\sum H : H \subseteq \{0,1\}^n_k\}$ of degree sequences of k-hypergraphs on $[n]$ with unrestricted numbers of edges. For $k = 2$, that is, graphs, these polytopes have been extensively studied; see [21] and the references therein. The Erdős–Gallai theorem implies that D^n_2 is the set of points $d \in \mathbb{R}^n$ satisfying the system

$$\sum_{i \in S} d_i - \sum_{i \in T} d_i \;\leq\; |S|(n-1-|T|)\;, \quad S, T \subseteq [n]\,,\; S \cap T = \emptyset\;,$$

and the vertices of D^n_2 were characterized in [13] as precisely the degree sequences of threshold graphs. More recently, the polytopes D^n_k for $k \geq 3$ were studied in [12, 17, 19], but neither a complete inequality description nor a complete characterization of vertices is known.

We now go back to the situation when the number of edges m is prescribed. We define the *degree sequence polytope* $D_k^{n,m}$ as the convex hull $D_k^{n,m} := \text{conv}\{\sum H : H \subseteq \{0,1\}_k^n\,,\ |H| = m\}$ of degree sequences of k-hypergraphs with m edges over $[n]$. A degree sequence is called *extremal* if it is a vertex of this polytope. We note that when maximizing a convex function over degree sequences there will always be an optimal solution which is extremal.

Again we restrict our attention to graphs and use alternatively the interpretation of a graph as $G \subseteq \{0,1\}_2^n$ and $G = ([n], E)$. The graph G is called a *threshold graph* if for some permutation π of $[n]$, each vertex $\pi(i)$ is connected to either all or none of the vertices $\pi(j)$, $j < i$. It is well known that a graph G with degree sequence $d = \sum G$ is threshold if and only if $N(i) \subseteq N[j]$ whenever $d_i \le d_j$, where $N(i) := \{j : \{i,j\} \in E\}$ and $N[i] := N(i) \uplus \{i\}$ are the open and closed neighborhoods of i, respectively; see [18]. We now characterize the extremal degree sequences of graphs with a prescribed number of edges in analogy with the result of [13].

THEOREM 6.2. *The vertices of $D_2^{n,m}$ are the degree sequences of threshold graphs with m edges. So for any convex function $f : \{0,1,\dots,m\}^n \to \mathbb{R}$ there is a threshold graph G maximizing $f(\sum G)$ and providing an optimal solution to the optimization problem over degree sequences.*

Proof. Suppose first that d is the degree sequence of a threshold graph with m edges. Then, by the result of [13], it is a vertex of D_2^n. Since $d \in D_2^{n,m} \subseteq D_2^n$, it is also a vertex of $D_2^{n,m}$.

Conversely, consider any vertex d of $D_2^{n,m}$. Let $w \in \mathbb{R}^n$ be such that wx is uniquely maximized over $D_2^{n,m}$ at d. Perturbing w if necessary we may assume that $w_1, \dots, w_n$ are distinct. Let $G = ([n], E)$ be a graph with $d = \sum G$. We need to show that $N(i) \subseteq N[j]$ whenever $d_i \le d_j$, which will imply that G is a threshold graph. Suppose on the contrary that for some $i \ne j$ we have $d_i \le d_j$ but there is some $k \in N(i) \setminus N[j]$.

Suppose first that $d_i = d_j$ and $w_i > w_j$. Then there must be also some $l \in N(j) \setminus N[i]$. Then the graph G' obtained from G by adding edge $\{i,l\}$ and dropping edge $\{j,l\}$ has m edges and $w\sum G' = w\sum G + w_i - w_j > wd$, which is impossible.

Next, consider the case where $d_i \le d_j$ and $w_j > w_i$. Then the graph G'' obtained from G by adding $\{j,k\}$ and dropping $\{i,k\}$ has m edges and $w\sum G'' = w\sum G + w_j - w_i > wd$, which is again impossible.

Thus, it remains to consider the case where $d_i < d_j$ and $w_i > w_j$. Let $d' \in D_2^{n,m}$ be the degree sequence obtained from d by setting $d'_i := d_j$, $d'_j := d_i$, and $d'_t := d_t$ for $t \ne i,j$. Then $wd' - wd = (w_i - w_j)(d_j - d_i) > 0$, which is a contradiction.

The second statement of the theorem now also follows since any convex function attains its maximum over a polytope at a vertex, which is the degree sequence of a threshold graph. ∎

Two remarks are in order here. First, we note that if the function is not convex, then, even if it is separable, it may be that no threshold graph is a maximizer. To see this, let $m = n$ and define univariate functions $f_1 = \cdots = f_n$ by $f_i(2) = 1$ and $f_i(t) = 0$ for all $t \ne 2$. Then any graph G with m edges whose degree sequence $d = \sum G$ maximizes $\sum_{i=1}^n f_i(d_i)$ must have a degree sequence satisfying $d_1 = \cdots = d_n = 2$ and so must be a (vertex) disjoint union of circuits. For all $n \ge 4$ any such graph is not threshold since by definition any threshold graph contains either an isolated vertex (of degree 0) or a dominating vertex (of degree $n - 1 \ge 3$).

Second, given a convex function f, we do not know how to efficiently find a vertex d of $D_2^{n,m}$ which maximizes f. Note that if we could find such a vertex d, then, since Theorem 6.2 guarantees that d must be the degree sequence of some threshold graph, we could also find an optimal threshold graph G with $\sum G = d$. Indeed, there must be some i with either $d_i = 0$ or $d_i = n - 1$. We then define $V' := [n] \setminus \{i\}$. If $d_i = 0$, then we recursively find a threshold graph G' on V' with degree sequence d' defined by $d'_i := d_i$ for all $i \in V'$ and add an isolated vertex i. If $d_i = n - 1$, we recursively find a threshold graph G' on V' with degree sequence d' defined by $d'_i := d_i - 1$ for all $i \in V'$ and add a dominating vertex i.

We conclude this section by showing that we can use Theorem 6.2 to obtain a dynamic programming algorithm for the optimization problem over degree sequences for any identical univariate functions $f_1 = \cdots = f_n = f$ which are convex. Observe that the time complexity that is established in the next theorem is significantly lower than the one established for the general case of (not necessarily convex) identical univariate functions in Theorem 1.2.

Theorem 6.3. *For $k = 2$, that is, graphs, optimization over degree sequences can be done in $O(n^2 m)$ time for any n, m and any identical univariate convex functions $f_1 = \cdots = f_n$.*

Proof. We use the fact that the functions $f_1 = \cdots = f_n = f$ are identical to assume that an optimal solution of our problem (which can be assumed to be a threshold graph by Theorem 6.2) satisfies that every vertex i is either adjacent to all vertices j such that $j > i$ (in which case we say that i is *dominating*) or it is not adjacent to any of those vertices (in which case we say that i is *isolating*). Thus, the terms dominating and isolating refer to the induced subgraph over the vertices with indices at least i.

Once again we use dynamic programming to find an optimal threshold graph for our problem. There are n stages, and in stage i we decide whether vertex i is dominating or isolating. A state in this dynamic program is a triple consisting of (i, e_i, δ_i) where e_i is the total number of edges adjacent to at least one vertex in $\{1, 2, \ldots, i\}$, and δ_i is the number of dominating vertices among $1, 2, \ldots, i$. The decision that $i+1$ is dominating means that its degree will be $\delta_i + n - (i+1)$, as $i+1$ will be adjacent to all dominating vertices in $1, \ldots, i$ and to all vertices in $i+2, \ldots, n$, and thus the reward of such a decision will be $f(\delta_i + n - (i+1))$, and in this case we move to state $(i+1, \delta_i + 1, e_i + n - (i+1))$. The decision that $i+1$ is isolating means that its degree will be δ_i, and thus the reward of such a decision will be $f(\delta_i)$ and we will move to the state $(i+1, \delta_i, e_i)$.

These decisions are feasible only when the third component is at most m (and the second component is at most n), and otherwise we have only the option of deciding that the vertex is isolated.

In the resulting dynamic programming we look for a maximum total reward path that leads from the initial state of $(0, 0, 0)$ to any of the final states defined as (n, δ, m) (choosing the value of δ that maximizes the total reward of the path).

The time complexity of this algorithm is $O(n^2 m)$, as there are $O(n^2 m)$ states (since there are $O(n)$ options for i, $O(n)$ options for δ_i, and $O(m)$ options for e_i) and the amount of work for each state is $O(1)$. ∎

We conclude this section with the following proposition on the complexity of deciding membership in degree sequence polytopes and deciding being extremal for these polytopes.

PROPOSITION 6.4. *It is polynomial time decidable, given n, k, m and rational $d \in \mathbb{R}^n$, whether or not d lies in the degree sequence polytope $D_k^{n,m}$ and whether or not d is a vertex of $D_k^{n,m}$.*

Proof. By the polynomial time equivalence of optimization and separation via the ellipsoid method, if we can do linear optimization over a rational polytope $P \subset \mathbb{R}^n$ in polynomial time, then we can also decide in polynomial time whether or not a given $d \in \mathbb{R}^n$ lies in P and whether or not d is a vertex of P; see [8] for more details. Since by Proposition 3.1 we can do linear optimization over $D_k^{n,m}$ for any n, k, m in polynomial time, the proposition follows. ∎

This proposition indicates that for any n, k, m, effective characterizations of inequalities defining $D_k^{n,m}$ and vertices of $D_k^{n,m}$ are plausible; these remain challenging research problems.

7. Shifted combinatorial optimization. Optimization over degree sequences can be viewed as a special case of the broad framework of *shifted combinatorial optimization* recently introduced and investigated in [6, 11, 14, 16].

Standard combinatorial optimization is the following extensively studied problem (see [23] for a detailed account of the literature and bibliography of thousands of articles on this).

(Standard) combinatorial optimization. Given $S \subseteq \{0,1\}^n$ and $w \in \mathbb{R}^n$, solve

$$\max\{ws \ : \ s \in S\} \ . \tag{1}$$

The complexity of this problem depends on the type and presentation of the defining set S.

Shifted combinatorial optimization is a broad nonlinear extension of this problem, which involves the choice of several feasible solutions from S at a time, defined as follows.

For a set $S \subseteq \mathbb{R}^n$ let S^m be the set of $n \times m$ matrices x having each column x^j in S. Call matrices $x, y \in \mathbb{R}^{n\times m}$ *equivalent* and write $x \sim y$ if each row of x is a permutation of the corresponding row of y. The *shift* of $x \in \mathbb{R}^{n\times m}$ is the unique matrix $\overline{x} \in \mathbb{R}^{n\times m}$ satisfying $\overline{x} \sim x$ and $\overline{x}^1 \geq \cdots \geq \overline{x}^m$. We can then define the following broad optimization framework.

Shifted combinatorial optimization. Given $S \subseteq \{0,1\}^n$ and $c \in \mathbb{R}^{n\times m}$, solve

$$\max\{c\overline{x} \ : \ x \in S^m\} \ . \tag{2}$$

This framework has a very broad expressive power, and its polynomial time solvability was so far established in the following situations:

- when S is presented by totally unimodular inequalities, in particular when S is the set of source-sink dipaths in a digraph or matchings in a bipartite graph [11];
- when S is the set of independent sets in a matroid, in particular spanning trees in a graph, or the intersection of two so-called strongly base-orderable matroids [16];
- when S is any property defined by any bounded monadic-second-order-logic formula over any graph of bounded tree-width [6].

Finally, a study of approximation algorithms for this and the closely related *separable multichoice optimization* framework over monotone systems was very recently undertaken in [14].

We claim that the optimization problem over degree sequences in multihypergraphs can be formulated as the special shifted combinatorial optimization problem with $S = \{0,1\}^n_k$. Then multihypergraphs are matrices $x \in S^m = (\{0,1\}^n_k)^m$ with degree sequence $\sum x = \sum_{j=1}^m x^j$.

Now, given functions $f_i : \{0,1,\ldots,m\} \to \mathbb{R}$, define $c \in \mathbb{R}^{n\times m}$ by $c_{i,j} := f_i(j) - f_i(j-1)$ for all i and j. Then for every $x \in S^m$ we have $c\overline{x} = \sum_{i=1}^n f_i(d_i) - \sum_{i=1}^n f_i(0)$, where $d = \sum x$. Therefore the multihypergraph x maximizes $c\overline{x}$ if and only if it maximizes $\sum_{i=1}^n f_i(d_i)$.

Next, for $S \subseteq \{0,1\}^n$ and positive integers m, u, let S^m_u be the set of matrices x in S^m such that for each $s \in S$ there are at most u columns of x which are equal to s. The *bounded* shifted combinatorial optimization problem is then $\max\{c\overline{x} : x \in S^m_u\}$. Then, defining c from the functions f_i as above, the degree sequence problem for hypergraphs can be formulated as a bounded shifted combinatorial optimization problem with $S = \{0,1\}^n_k$ as before, and $u = 1$.

Thus, our results in this paper for hypergraphs can be regarded as a first step in the development of a theory of bounded shifted combinatorial optimization.

REFERENCES

[1] D. Billington, *Degree Multisets of Hypergraphs*, Ph.D. thesis, University of Melbourne, 1982.

[2] D. Billington, *Conditions for degree sequences to be realisable by 3-uniform hypergraphs*, J. Combin. Math. Combin. Comput., 3 (1988), pp. 71–91.

[3] C. J. Colbourn, W. L. Kocay, and D. R. Stinson, *Some NP-complete problems for hypergraph degree sequences*, Discrete Appl. Math., 14 (1986), pp. 239–254.

[4] P. Erdős and T. Gallai, *Graphs with prescribed degrees of vertices*, Mat. Lopak, 11 (1960), pp. 264–274 (in Hungarian).

[5] P. Erdős, Z. Király, and I. Miklós, *On the swap-distances of different realizations of a graphical degree sequence*, Combin. Probab. Comput., 22 (2013), pp. 366–383.

[6] J. Gajarský, P. Hliněný, M. Koutecký, and S. Onn, *Parameterized shifted combinatorial optimization*, in Computing and Combinatorics, Lecture Notes in Comput. Sci. 10392, Springer, 2017, pp. 224–236.

[7] M. R. Garey and D. S. Johnson, *Computers and Intractability*, Freeman, 1979.

[8] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, 1993.

[9] S. L. Hakimi, *On realizability of a set of integers as degrees of the vertices of a linear graph.* I., J. Soc. Indust. Appl. Math., 10 (1962), pp. 496–506.

[10] V. Havel, *A remark on the existence of finite graphs*, Časopis Pěst. Mat., 80 (1955), pp. 477–480.

[11] V. Kaibel, S. Onn, and P. Sarrabezolles, *The unimodular intersection problem*, Oper. Res. Lett., 43 (2015), pp. 592–594.

[12] C. Klivans and V. Reiner, *Shifted set families, degree sequences, and plethysm*, Electron. J. Combin., 15 (2008), 14.

[13] M. Koren, *Extreme degree sequences of simple graphs*, J. Combinatorial Theory Ser. B, 15 (1973), pp. 213–224.

[14] M. Koutecký, A. Levin, S. M. Meesum, and S. Onn, *Approximate separable multichoice optimization over monotone systems*, submitted.

[15] E. L. Lawler, *A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem*, Management Sci., 18 (1971/72), pp. 401–405.

[16] A. Levin and S. Onn, *Shifted matroid optimization*, Oper. Res. Lett., 44 (2016), pp. 535–539.

[17] R. I. Liu, *Nonconvexity of the set of hypergraph degree sequences*, Electron. J. Combin., 20 (2013), no. 1, 21.

[18] N. V. R. Mahadev and U. N Peled, *Threshold Graphs and Related Topics*, Ann. Discrete Math. 56, North–Holland, 1995.

[19] N. L. B. Murthy and M. K. Srinivasan, *The polytope of degree sequences of hypergraphs*, Linear Algebra Appl., 350 (2002), pp. 147–170.

[20] U. N. Peled, R. Petreschi, and A. Sterbini, *(n,e)-graphs with maximum sum of squares of*

degrees, J. Graph Theory, 31 (1999), pp. 283–295.

[21] U. N. Peled and M. K. Srinivasan, *The polytope of degree sequences*, Linear Algebra Appl., 114/115 (1989), pp. 349–377.

[22] H. J. Ryser, *Combinatorial properties of matrices of zeroes and ones*, Canad. J. Math., 9 (1957), pp. 371–377.

[23] A. Schrijver, *Combinatorial Optimization*, Springer, 2003.

[24] D. B. West, *Introduction to Graph Theory*, Prentice–Hall, 1996.