McMaster University

Advanced Optimization Laboratory

Title:

Optimization over degree sequences

Authors:

Antoine Deza, Asaf Levin, Syed M. Meesum, and Shmuel Onn

AdvOL-Report No. 2017/5

July 2017, Hamilton, Ontario, Canada

Optimization over Degree Sequences

Antoine Deza [*] Asaf Levin [†] Syed M. Meesum [‡] 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 if a given sequence is the degree sequence of a 3-hypergraph is computationally prohibitive, thereby solving a 30 year long open problem. This implies that optimization over hypergraphs is hard already 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.

Keywords: graph, hypergraph, combinatorial optimization, degree sequence, threshold graph, shifted combinatorial optimization

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 starting with the celebrated Erdös-Gallai paper [2, 1960] which effectively characterizes the degree sequences of graphs.

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 before, see [15].

More generally, we are interested in the problem over (uniform) k-hypergraphs. In the sequel 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 are also interested in the following. 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 a $0-1$ vector containing exactly k ones), so that

[*]McMaster University, Hamilton. Email: deza@mcmaster.ca
[†]Technion - Israel Institute of Technology, Haifa. Email: levinas@ie.technion.ac.il.
[‡]Institute of Mathematical Sciences, HBNI, Chennai. Email: meesum@imsc.res.in
[§]Technion - Israel Institute of Technology, Haifa. Email: onn@ie.technion.ac.il

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 functions $f_i : \{0, \ldots, m\} \to \mathbb{R}$, find a k-(multi)hypergraph H whose degree sequence $d := \sum H$ maximizes $\sum_{i=1}^{n} f_i(d_i)$.

The case of linear functions is very simple and will be discussed in the short 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* $f_1, \ldots, f_n$.

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* $O(n^5 m^2)$ *time for any* n, m *and any identical univariate functions* $f_1 = \cdots = f_n$.

When the identical univariate functions are *convex* we give a better bound in Theorem 6.3.

In order to obtain our result on the negative side, we consider also the 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 Erdös-Gallai theorem [2, 1960] 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-Stinson [1, Problem 3.1, 1986] and remained open to date. We solve it in Section 2 by showing the following theorem.

Theorem 1.3 *It is NP-complete to decide if* $d \in \mathbb{Z}_+^n$ *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 already 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 $f : \{0, 1, \ldots, m\}^n \to \mathbb{R}$ which is not necessarily separable as considered above. For this we discuss the *degree sequence polytopes* studied in [7, 8, 12, 14, 16] and references therein, introduce and study degree sequence polytopes of hypergraphs with prescribed number of edges, and show that for $k = 2$ their vertices correspond to suitable *threshold graphs* [13].

Finally, in Section 7, we show that optimization over degree sequences can be viewed within the framework of *shifted combinatorial optimization* recently introduced and investigated in the series of papers [3, 6, 9, 11], 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 and prove the following theorem solving a problem raised 30 years ago by Colbourn-Kocay-Stinson [1, Problem 3.1, 1986].

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

Proof. The problem is in NP since if d is a degree sequences 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 if 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 [4] and we reduce it to ours.

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 $\sum H = \mathbf{1}$. Note that if this holds then $w\mathbf{1} = w\sum H = \sum\{wx : x \in H\} = 0$ so we may and do assume $w\mathbf{1} = 0$ (that is, 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).

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

Define $d := \mathbf{1} + \sum S_+$. 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 the latter is NP-complete.

We now prove the claim. Suppose first H satisfies $wx = 0$ for all $x \in H$ and $\sum H = \mathbf{1}$. Then $H \subseteq S_0$ 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

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

with equality if and only if $G \cap S_- = \emptyset$ and $G \cap S_+ = S_+$, since $\mathrm{sign}(wx) = \sigma$ 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. ◻

The last hardness result for 3-hypergraph is easily extended for k-hypergraph for all fixed values of k such that $k \geq 3$ as we now demonstrate.

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

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

Next we reduce the problem for 3-hypergraph into the problem for k-hypergraph. Let $d = (d_1, d_2, \ldots, d_n)$ be the instance of the input sequence for the problem on 3-hypergraph. Let $m = \frac{1}{3}\sum_{i=1}^n d_i$ be the number of edges, and the question is equivalent to the question if d

is a degree sequence of a 3-hypergraph with m edges. We construct an input to the problem for k-hypergraphs by letting $d' = (d'_1, d'_2, \dots d'_{n+k-3})$ where $d'_i = d_i$ for $1 \le i \le n$, and $d'_i = m$ for $i > n$. The problem is to decide if d' is a degree sequence of a k-hypergraph with m edges.

The correctness of this reduction follows as any such k-hypergraph must satisfy that every edge must contain the vertices $n+1, \dots, n+k-3$. Thus, if d is a degree sequence of a 3-hypergraph H, we take each edge in H and add $n+1, \dots n+k-3$ to obtain an edge on $[n+k-3]$, and the collection of these edges is a k-hypergraph with degree sequence d'. In the other direction, if d' is a degree sequence of k-hypergraph H' on $[n+k-3]$, then every edge of H' must contain the vertices $n+1, \dots n+k-3$, and if we remove these vertices from each edge of H' we get a collection of edges in $[n]$ and the corresponding 3-hypergraph has degree sequence d. ◻

3 Linear functions

Here we discuss the simple 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

$$\max\{w \sum H \ : \ H \in (\{0,1\}^n_k)^m\} \ , \qquad \max\{w \sum H \ : \ H \subseteq \{0,1\}^n_k \, , \ |H| = m\} \ .$$

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 $x \in \{0,1\}^n_k$ maximizing wx. Such an x can be found by sorting w, that is, if for a permutation π of $[n]$ we have $w_{\pi(1)} \ge \cdots \ge w_{\pi(n)}$, 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 $x \in \{0,1\}^n_k$ with largest values wx. For k fixed we can simply compute wx for each of the $O(n^k)$ vectors in $\{0,1\}^n_k$ in polynomial time and pick the best m vectors. For variable k, we can use the algorithm of Lawler [10] to find the m vectors $x \in \{0,1\}^n_k$ with largest values wx in time polynomial in n, k, m. ◻

4 Arbitrary functions over multihypergraphs

For multihypergraphs there is a simple characterization of degree sequences which follows from results of [17, 1957] on $0-1$ matrices, and a simple greedy procedure for constructing a hypergraph from its degree sequence. We record these facts below and provide a short proof.

Proposition 4.1 *A 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 $d_i \le m$ for all i, and H is efficiently constructible 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 $d_k \geq 1$ and $d_{k+1} \leq m - 1$. If $d_k = 0$ then $km = \sum d_i \leq (k-1)m$ which is impossible. If $d_{k+1} = m$ 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 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,

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

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

Finding an optimal d can be done by 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 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 [5, 1962] which leads to a simple greedy procedure for constructing a graph from its degree sequence. A vector $d \in \mathbb{Z}_+^n$ is *reducible* if, permuting it so 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 $d' := (d_2 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n)$. Here is the statement and the short proof.

Proposition 5.1 *A vector $d \in \mathbb{Z}_+^n$ is a degree sequence of a graph G if and only if d is reducible and its reduction is also a degree sequence. So G is recursively constructible 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 recursively constructible 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 sequences of $G = ([n], E)$. We show 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 Erdös-Gallai characterization of sorted degree sequences, we need to solve the following nonlinear integer optimization problem,

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

find an optimal d, and then use the algorithm of Proposition 5.1 to find a 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 $\beta_i = i$ and 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 Erdös-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 $1, 2, ..., i$ 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 $f(d_{i+1}) + \beta_{i+1} \cdot f(i+1)$. If such a decision is feasible then it will results in a transition to the state $(i+1, p_i + d_{i+1}, d_{i+1}, \beta_{i+1}, s_i + (i+1) \cdot \beta_{i+1})$.

It remains to define the set of pairs (d_{i+1}, β_{i+1}) of non-negative 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 means that the degree of a vertex will be defined at most once (it 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).

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. $p_i + d_{i+1} + s_i + (i+1) \cdot \beta_{i+1} \leq m$ - this condition will enforce that we can choose at most m edges.

4. The $(i+1)$-th Erdös-Gallai constraint that can be stated as follows
$$p_i + d_{i+1} - (i+1) \cdot i \leq s_i + (i+1) \cdot \beta_{i+1} + (\beta_{i+1} - i - 1) \cdot (i+1).$$

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 Erdös-Gallai constraint. This is so as the left hand side will not change when we determine the degrees in the next stages, and the right hand side will also not changed as all the degrees that we select in the following stages will be between $i+2$ and d_{i+1} so the minimum on the right hand side for the indices between $i+2$ and β_{i+1} will be $i+1$ as we assume 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 $O(n^3m^2)$ 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 which turns out to be quite simple.

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 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 number of edges. For $k = 2$, that is, graphs, these polytopes have been extensively studied, see [16] and the references therein. The Erdös-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 [8] as precisely the degree sequences of threshold graphs. More recently, the polytopes D^n_k for $k \geq 3$ were studied in [7, 12, 14], 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* $D^{n,m}_k$ as the convex hull $D^{n,m}_k := \mathrm{conv}\{\sum H : H \subseteq \{0,1\}^n_k \,,\, |H| = m\}$ of degree sequence 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 attention to graphs and use alternatively the interpretation of a graph as $G \subseteq \{0,1\}^n_2$ 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 $d_i \leq d_j$, 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 [13]. We now characterize the extremal degree sequences of graphs with prescribed number of edges in analogy with the result of [8].

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* $f : \{0, 1, \ldots, m\}^n \to \mathbb{R}$ *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 [8], 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 $w_1, \ldots, w_n$ are distinct. Let $G = ([n], E)$ be a graph with $d = \sum G$. We need to show that $N(i) \subseteq N[j]$ whenever $d_i \leq d_j$, which will imply that G is a threshold graph. Suppose on the contrary that for some $i \neq j$ we have $d_i \leq d_j$ but there is some $k \in N(i) \setminus N[j]$.

Suppose first $d_i = d_j$ 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 $d_i \leq d_j$ 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 impossible again.

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 $t \neq i, j$. 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 $t \neq 2$. Then any graph G with m edges whose degree sequence $d = \sum G$ maximizes $\sum_{i=1}^n f_i(d_i)$ must have degree sequence satisfying $d_1 = \cdots = d_n = 2$ and so must be a (vertex) disjoint union of circuits. For all $n \geq 4$ 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 $n - 1 \geq 3$).

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 the above 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 $O(n^2m)$ 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 if vertex i is dominating or isolating. A state in this dynamic programming 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 one 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 $O(n^2m)$ as there are $O(n^2m)$ 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 $O(1)$. □

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 [3, 6, 9, 11].

Standard combinatorial optimization is the following extensively studied problem (see [18] 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. First, when S is presented by totally unimodular inequalities, in particular when S is the set of matchings in a bipartite graph [6]. Second, when S is the set of independent sets in a matroid or the intersection of two so-called strongly-base-orderable matroids [11]. Third, when S is any property defined by any bounded monadic-second-order-logic formula over any graph of bounded tree-width [3]. And finally, a study of approximation algorithms for this framework was very recently initiated in [9].

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.

Acknowledgment

A. Deza was partially supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant programs (RGPIN-2015-06163). A. Levin was partially supported by a Grant from GIF - the German-Israeli Foundation for Scientific Research and Development (grant number I-1366-407.6/2016). S.M. Meesum was partially supported by a grant at the Technion. S. Onn was partially supported by the Dresner Chair at the Technion.

References

[1] Colbourn, C.J., W.L. Kocay, W.L., Stinson, D.R.: Some NP-complete problmes for hypergraph degree sequences. Discrete Applied Mathematics 14:239–254 (1986)

[2] Erdos, P., Gallai, T.: Graphs with prescribed degrees of vertices (in Hungarian). Matematikai Lopak 11:264–274 (1960)

[3] Gajarský J., Hliněný P., Koutecký M., Onn S.: Parameterized shifted combinatorial optimization. Proceedings of the 23rd Annual International Computing and Combinatorics Conference (2017), Lecture Notes in Computer Science, Springer, to appear.

[4] Garey, M.R., Johnson, D.S.: Computers and Intractability. (1979), Freeman

[5] Hakimi, S.L.: On realizability of a set of integers as degrees of the vertices of a linear graph. SIAM Journal 10:496–506 (1962)

[6] Kaibel V., Onn S., Sarrabezolles P.: The unimodular intersection problem. Operations Research Letters 43:592–594 (2015)

[7] Klivans, C., Reiner, V.: Shifted set families, degree sequences, and plethysm. The Electronic Journal of Combinatorics 15:#R14 (2008)

[8] Koren, M.: Extreme degree sequences of simple graphs. Journal of Combinatorial Theory Series B 15:213–224 (1973)

[9] Koutecký M., Levin, A., Meesum S.M., Onn S.: Approximate shifted combinatorial optimization. Submitted.

[10] Lawler, E.L.: A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. Management Science 18:401–405 (1972)

[11] Levin A., Onn S.: Shifted matroid optimization. Operations Research Letters 44:535–539 (2016)

[12] Liu, R.I.: Nonconvexity of the set of hypergraph degree sequences. The Electronic Journal of Combinatorics 20:#P21 (2013)

[13] Mahadev, N.V.R., Peled, U.N.: Threshold Graphs and Related Topics. Annals of Discrete Mathematics 56, North-Holland (1995)

[14] Murthy, N.L.B., Srinivasan, M.K.: The polytope of degree sequences of hypergraphs. Linear Algebra and its Applications 350:147–170 (2002)

[15] Peled, U.N., Petreschi, R., Sterbini, A.: (n, e)-graphs with maximum sum of squares of degrees. Journal of Graph Theory 31:283–295 (1999)

[16] Peled, U.N., Srinivasan, M.K.: The polytope of degree sequences. Linear Algebra and its Applications 114/115:349–377 (1989)

[17] Ryser, H.J.: Combinatorial properties of matrices of zeroes and ones. Canadian Journal of Mathematics 9:371–377 (1957)

[18] Schrijver A.: Combinatorial Optimization. Springer (2003)