Degree-Bounded Generalized Polymatroids and Approximating the Metric Many-Visits TSP*

Kristóf Bérczi[†]   André Berger[‡]   Matthias Mnich[§]   Roland Vincze[¶]

Abstract

In the Bounded Degree Matroid Basis Problem, we are given a matroid and a hypergraph on the same ground set, together with costs for the elements of that set as well as lower and upper bounds $f(\varepsilon)$ and $g(\varepsilon)$ for each hyperedge ε. The objective is to find a minimum-cost basis B such that $f(\varepsilon) \leqslant |B \cap \varepsilon| \leqslant g(\varepsilon)$ for each hyperedge ε. Király et al. (Combinatorica, 2012) provided an algorithm that finds a basis of cost at most the optimum value which violates the lower and upper bounds by at most $2\Delta - 1$, where Δ is the maximum degree of the hypergraph. When only lower or only upper bounds are present for each hyperedge, this additive error is decreased to $\Delta - 1$.

We consider an extension of the matroid basis problem to generalized polymatroids, or g-polymatroids, and additionally allow element multiplicities. The Bounded Degree g-polymatroid Element Problem with Multiplicities takes as input a g-polymatroid $Q(p, b)$ instead of a matroid, and besides the lower and upper bounds, each hyperedge ε has element multiplicities m_ε. Building on the approach of Király et al., we provide an algorithm for finding a solution of cost at most the optimum value, having the same additive approximation guarantee.

As an application, we develop a 1.5-approximation for the metric Many-Visits TSP, where the goal is to find a minimum-cost tour that visits each city v a positive $r(v)$ number of times. Our approach combines our algorithm for the Bounded Degree g-polymatroid Element Problem with Multiplicities with the principle of Christofides' algorithm from 1976 for the (single-visit) metric TSP, whose approximation guarantee it matches.

Keywords: Generalized polymatroids, degree constraints, traveling salesman problem.

arXiv:1911.09890v2 [cs.DM] 14 Dec 2019

1 Introduction

In this paper we consider polymatroidal optimization problems with degree constraints. An illustrious example is the MINIMUM BOUNDED DEGREE SPANNING TREE problem, where the goal is to find a minimum cost spanning tree in a graph with lower and upper bounds on the degree at each vertex. Checking feasibility of a degree-bounded spanning tree contains the NP-hard Hamiltonian path problem; therefore, efficiently finding spanning trees that only slightly violate the degree constraints, is of interest. Several algorithms were given that were balancing between the cost of the spanning tree and the violation of the degree bounds [5, 6, 15, 24, 25]. Goemans [16] gave a polynomial-time algorithm that finds a spanning tree of cost at most the optimum value that violates each degree bound by at most 2. Singh and Lau [31] improved the additive approximation guarantee to 1 by extending the iterative rounding method of Jain [22] with a relaxation step. Zenklusen [34] considered an extension of the problem where for every vertex v, the edges adjacent to v have to be independent in a given matroid.

Motivated by a problem on binary matroids posed by Frienze, a matroidal generalization called the MINIMUM BOUNDED DEGREE MATROID BASIS PROBLEM was introduced by Király, Lau and Singh [23] in 2012. The problem takes as input a matroid $M = (S, r)$, a cost function $c : S \to \mathbb{R}$, a hypergraph $H = (S, \mathcal{E})$ and lower and upper bounds $f, g : \mathcal{E} \to \mathbb{Z}_{\geqslant 0}$; the objective is to find a minimum-cost basis B of M such that $f(\varepsilon) \leqslant |B \cap \varepsilon| \leqslant g(\varepsilon)$ for each $\varepsilon \in \mathcal{E}$. For this problem, the authors developed an approximation algorithm that is based on the iterative relaxation method and a clever token-counting argument of Chaudhuri et al. [5] and Singh and Lau [31]. Let us denote the maximum degree of the hypergraph H by Δ. When both lower bounds and upper bounds are present, their algorithm returns a basis B of cost at most the optimum value such that $f(\varepsilon) - 2\Delta + 1 \leqslant |B \cap \varepsilon| \leqslant g(\varepsilon) + 2\Delta - 1$ holds for each $\varepsilon \in \mathcal{E}$. Based on a technique of Bansal et al. [2], they showed that the additive error can be improved when only lower bounds (or only upper bounds) are present, thus finding a basis of cost B at most the optimum value such that $|B \cap \varepsilon| \leqslant g(\varepsilon) + \Delta - 1$ (respectively, $f(\varepsilon) - \Delta + 1 \leqslant |B \cap \varepsilon|$) for each $\varepsilon \in \mathcal{E}$. Bansal et al. [1] considered extensions of the MINIMUM BOUNDED DEGREE MATROID BASIS PROBLEM to contra-polymatroid intersection and to crossing lattice polyhedra. In all of these cases, the solution for the problem is a $0-1$ vector defined on the ground set.

Our results

In this paper we consider a different generalization of the BOUNDED DEGREE MATROID BASIS PROBLEM. The generalization deals with general polymatroids (or g-polymatroids) instead of matroids, and additionally allows multiplicities of the hyperedges. Formally, the problem takes as input a g-polymatroid $Q(p, b) = (S, p, b)$ with a cost function $c : S \to \mathbb{R}$, and a hypergraph $H = (S, \mathcal{E})$ on the same ground set with lower and upper bounds $f, g : \mathcal{E} \to \mathbb{Z}_{\geqslant 0}$ and multiplicity vectors $m_\varepsilon : S \to \mathbb{Z}_{>0}$ for $\varepsilon \in \mathcal{E}$ satisfying $m_\varepsilon(s) = 0$ for $s \in S - \varepsilon$. The objective is to find a minimum-cost element x of $Q(p, b)$ such that $f(\varepsilon) \leqslant \sum_{s \in \varepsilon} m_\varepsilon(s)x(s) \leqslant g(\varepsilon)$ for each $\varepsilon \in \mathcal{E}$. We call this problem the BOUNDED DEGREE g-POLYMATROID ELEMENT PROBLEM WITH MULTIPLICITIES.

Our first main algorithmic result is the following:

Theorem 1. *There is a polynomial-time algorithm for the* BOUNDED DEGREE g-POLYMATROID ELEMENT PROBLEM WITH MULTIPLICITIES *which returns an element x of $Q(p, b)$ of cost at most the optimum value such that $f(\varepsilon) - 2\Delta + 1 \leqslant \sum_{s \in \varepsilon} m_\varepsilon(s)x(s) \leqslant g(\varepsilon) + 2\Delta - 1$ for each $\varepsilon \in \mathcal{E}$, where $\Delta = \max_{s \in S} \{\sum_{\varepsilon \in \mathcal{E} : s \in \varepsilon} m_\varepsilon(s)\}$.*

Theorem 1 extends the result of Király et al. [23] from matroids to g-polymatroids. It turns out that, when upper bounds are present, there is a significant difference when g-polymatroids

are considered instead of matroids. Adapting the algorithm of Király et al. is not immediate, as a crucial step of their approach is to relax the problem by deleting a constraint corresponding to a hyperedge ε with small $g(\varepsilon)$ value. This step is feasible when the solution is a 0-1 vector, as in those cases the violation on ε is upper bounded by the size of the hyperedge. This does not hold for g-polymatroids (or even for polymatroids), where an integral element might have coordinates larger than 1. However, we show that after the first round of our algorithm, the problem can be restricted to the unit cube and so upper bounds remain tractable.

When only lower bounds (or only upper bounds) are present, we call the problem LOWER (UPPER) BOUNDED DEGREE g-POLYMATROID ELEMENT PROBLEM WITH MULTIPLICITIES. In this case, we show a similar result with an improved additive error:

Theorem 2. *There is an algorithm for the* LOWER BOUNDED DEGREE g-POLYMATROID ELE-MENT PROBLEM WITH MULTIPLICITIES *that runs in polynomial time and returns an element x of $Q(p, b)$ of cost at most the optimum value such that $f(\varepsilon) - \Delta + 1 \leqslant \sum_{s \in \varepsilon} m_\varepsilon(s) x(s)$ for each $\varepsilon \in \mathcal{E}$. An analogous result holds for the* UPPER BOUNDED DEGREE g-POLYMATROID ELEMENT PROBLEM, *where $\sum_{s \in \varepsilon} m_\varepsilon(s) x(s) \leqslant g(\varepsilon) + \Delta - 1$.*

While being interesting by itself, the algorithm alluded to in Theorem 2 serves as the key ingredient for our second main algorithmic result. It concerns an extension of the TRAVELING SALESMAN PROBLEM (TSP), one of the cornerstones of combinatorial optimization. In TSP, we are given a set of n cities with their pairwise non-negative symmetric distances, and we seek a tour of minimum overall length that visits every city exactly once and returns to the origin. For the metric variant, when distances obey the triangle inequality, Christofides [7] in 1976 gave a polynomial-time algorithm that returns a 1.5-approximation to the optimal tour. The algorithm was independently discovered by Serdyukov [30]. For more than 40 years, no polynomial-time algorithm with better approximation guarantee has been discovered.

In the generalization of the TSP, known as the MANY-VISITS TSP, each city v is equipped with a request $r(v) \in \mathbb{Z}_{\geqslant 1}$, and we seek a tour of minimum overall length that visits city v exactly $r(v)$ times and returns to the origin. Note that a loop might have a positive cost at any city in this case. The MANY-VISITS TSP was first considered in 1966 by Rothkopf [28]. The problem is clearly NP-hard as it generalizes the TSP. In 1980, Psaraftis [27] gave a dynamic programming algorithm with time complexity $\mathcal{O}(n^2 \prod_{i=1}^{n} (r_i + 1))$; observe that this value may be as large as $(r/n + 1)^n$, which is prohibitive even for moderately large values of $r = \sum_{i=1}^{n} r_i$. In 1984, Cosmadakis and Papadimitriou [8] designed a family of algorithms, the fastest of which has run time[1] $\mathcal{O}^\star(n^{2n} 2^n + \log \sum r_i)$.

The analysis of the algorithm is highly non-trivial, combining graph-theoretic insights and involved estimates of various combinatorial quantities. The usefulness of the Cosmadakis-Papadimitriou algorithm is limited by its superexponential dependence on n in the run time, as well as its superexponential space requirement. Recently, Berger et al. [3] simultaneously improved the run time to $2^{\mathcal{O}(n)} \cdot \log \sum r_i$ and reduced the space complexity to polynomial.

As it is a generalization of the TSP, the MANY-VISITS TSP is of fundamental interest. This framework can be used for modeling *high-multiplicity* scheduling problems [4,20,27,33]. In such problems, every job belongs to a job type, and two jobs of the same type are considered to be identical. One notable example of such problems is the *aircraft sequencing problem*. Airplanes are categorized into a small number of different classes. Two airplanes belonging to the same class need the same amount of time to land. In addition, there is a minimum time that should pass between the arrival of two planes. The amount of this time only depends on the classes of the two airplanes, and the aim is to minimize the time when the last plane lands.

---

[1]The $\mathcal{O}^\star$ notation suppresses the factors polynomial in n.

At the Hausdorff Workshop on Combinatorial Optimization in 2018, Rico Zenklusen brought up the topic of approximation algorithms for the metric version of MANY-VISITS TSP in the context of iterative relaxation techniques; he suggested an approach to obtain a 1.5-approximation, which is unpublished. The cost function being metric implies that the cost of each loop c_{ii} is at most twice the cost of leaving city i to any other city j and returning. The assumption of metric costs is necessary, as the TSP, and therefore the MANY-VISITS TSP does not admit any non-trivial approximation for unrestricted cost functions.

Our next algorithmic result shows that a polynomial-time algorithm that matches the approximation guarantee of Christofides and Serdyukov for the single-visit case indeed exists.

Theorem 3. *There is a polynomial-time* 1.5-*approximation for the metric* MANY-VISITS TSP.

Let us remark that the requirements $r(v)$ are encoded in binary. The TSP can also be formulated for directed graphs, where the cost function is asymmetric. In a recent breakthrough, Svensson et al. [32] gave the first constant-factor approximation for the metric ATSP. We can show the following:

Theorem 4. *There is a polynomial-time* $\mathcal{O}(1)$-*approximation for the metric* MANY-VISITS ATSP.

The rest of the paper is organized as follows. In Sect. 2, we give an overview of the notation and definitions. In Sect. 3, we provide a simple 2.5-approximation for the metric MANY-VISITS TSP that runs in polynomial-time, and a polynomial-time constant-factor approximation for the metric MANY-VISITS ATSP. Thereafter, in Sect. 4, we give the necessary background on g-polymatroids. Sect. 5 describes the approximation algorithm for the BOUNDED DEGREE g-POLYMATROID ELEMENT PROBLEM WITH MULTIPLICITIES. The 1.5-approximation for the metric MANY-VISITS TSP is given in Sect. 6. We conclude in Sect. 7.

2 Preliminaries

Throughout the paper, we let $G = (V, E)$ be a finite, undirected complete graph on n vertices, whose edge set E also contains a self-loop at every vertex $v \in V$. For a subset $F \subseteq E$ of edges, the *set of vertices covered by F* is denoted by $V(F)$. The *number of connected components* of the graph $(V(F), F)$ is denoted by $\text{comp}(F)$. For a subset $X \subseteq V$ of vertices, the *set of edges spanned by X* is denoted by $E(X)$. The set of edges incident to a vertex v is denoted by $\delta(v)$. For a vector $x \in \mathbb{R}^{|E|}$, we denote the sum of the x-values on the edges incident to v by $d_x(v)$. Note that the x-value of the self-loop at v is counted twice in $d_x(v)$. Given two graphs H_1, H_2 on the same vertex set, $H_1 + H_2$ denotes the multigraph on the same vertex set obtained by taking the union of the edge sets of H_1 and H_2.

Given a vector $x \in \mathbb{R}^{|S|}$ and a set $Z \subseteq S$, we use $x(Z) = \sum_{s \in Z} x(s)$. The *lower integer part of x* is denoted by $\lfloor x \rfloor$, so $\lfloor x \rfloor(s) = \lfloor x(s) \rfloor$ for every $s \in S$. This notation extends to sets as well, therefore by $\lfloor x \rfloor(Z)$ we mean $\sum_{s \in Z} \lfloor x \rfloor(s)$. The *support of x* is denoted by $\text{supp}(x)$, that is, $\text{supp}(x) = \{s \in S : x(s) \neq 0\}$. The *difference of set B from set A* is denoted by $A - B = \{s \in A : s \notin B\}$. We denote a single-element set $\{s\}$ by s, and with a slight abuse of notation, will write $A - s$ to indicate $A - \{s\}$. The *charasteristic vector* of a set A is denoted by χ_A.

Let $\mathcal{T}$ be a collection of subsets of S. We call $\mathcal{L} \subseteq \mathcal{T}$ an *independent laminar system* if for any pair $X, Y \in \mathcal{L}$: (i) they do not intersect, i.e. either $X \subseteq Y$, $Y \subseteq X$ or $X \cap Y = \emptyset$, (ii) the characteristic vectors χ_Z of the sets $Z \in \mathcal{L}$ are independent. A *maximal* independent laminar system $\mathcal{L}$ with respect to $\mathcal{T}$ is an independent laminar system in $\mathcal{T}$, such that for any $Y \in \mathcal{T} - \mathcal{L}$

the system $\mathcal{L} \cup \{Y\}$ is not independent laminar. In other words, if we include any set Y from $\mathcal{T} - \mathcal{L}$, it will intersect at least one set Y from $\mathcal{L}$, or χ_Y can be given as a linear combination of $\{\chi_Z : Z \in \mathcal{L}\}$. Given a laminar system $\mathcal{L}$ and a set $X \subseteq S$, the set of maximal members of $\mathcal{L}$ lying inside X is denoted by $\mathcal{L}^{\max}(X)$, that is, $\mathcal{L}^{\max}(X) = \{Y \in \mathcal{L} : Y \subset X, \nexists Y' \in \mathcal{L} \text{ s.t. } Y \subset Y' \subset X\}$.

The cost functions $c : E \to \mathbb{R}_{\geqslant 0}$ are assumed to satisfy the triangle inequality. The *minimum cost of an edge incident to a vertex* v is denoted by $c_v^{\min} := \min_{u \in V} c(uv)$. Note that $u = v$ is allowed in the definition, therefore the minimum takes into account the cost of the self-loop at v as well. The triangle inequality holds for self-loops, too, meaning that $c(vv) \leqslant 2 \cdot c_v^{\min}$ for all $v \in V$.

In the MANY-VISITS TSP, each vertex $v \in V$ is additionally equipped with a request $r(v) \in \mathbb{Z}_{\geqslant 1}$ encoded in binary. The goal is to find a minimum-cost closed walk (or *tour*) on the edges of the graph that visits each vertex $v \in V$ exactly $r(v)$ times. Listing all the edges of such a walk might be exponential in the size of the input, hence we always consider *compact representations* of the solution and the multigraphs that arise in our algorithms. That is, rather than storing an $r(V)$-long sequence of edges, for every edge e we store its multiplicity $z(e)$ in the solution. As there are at most n^2 different edges in the solution each having multiplicity at most $\max_{v \in V} r(v)$, the space needed to store a feasible solution is $\mathcal{O}(n^2 \log r(V))$. Therefore a vector $z \in \mathbb{Z}_{\geqslant 0}^E$ represents a feasible tour if $d_z(v) = 2 \cdot r(v)$ for every $v \in V$ and $\operatorname{supp}(z)$ is a connected subgraph of G.

From this compact representation, one can compute a collection $\mathcal{C}$ of pairs (C, μ_C), where each C is a simple closed walk (cycle) and μ_C is the corresponding integer denoting the number of copies of C. The number of such cycles C is polynomial in n, and one can compute $\mathcal{C}$ in polynomial time (see, e.g., the procedure in Sect. 2 of Grigoriev and van de Klundert [17]). One can obtain the explicit order of the vertices from (C, μ_C) the following way: traverse μ_C copies of an arbitrary cycle C, and whenever a vertex u is reached for the first time, traverse $\mu_{C'}$ copies of every cycle $C' \neq C$ containing u. Note that while the size of $\mathcal{C}$ is polynomial in n, the size of the explicit order of the vertices is exponential, hence the time complexity of the last step is also exponential in n.

Denote by $\mathsf{T}_{c,r}^\star$ an optimal solution for an instance (G, c, r) of the MANY-VISITS TSP, and by $\mathsf{T}_{c,1}^\star$ an optimal tour for the single-visit TSP (i.e., when $r(v) = 1$ for each $v \in V$). Relaxing the connectivity requirement for solutions of the MANY-VISITS TSP yields Hitchcock's transportation problem, which is solvable in polynomial time [10] and whose optimal solution we denote by $\mathsf{TP}_{c,r}^\star$.

3   A Simple 2.5-Approximation for the Metric Many-Visits TSP

In this section we give a simple 2.5-approximation algorithm for the metric MANY-VISITS TSP; see Algorithm 1.

Algorithm 1 A polynomial-time $(\alpha + 1)$-approximation for the metric MANY-VISITS TSP.

> **Input:** A complete undirected graph G, costs $c : E \to \mathbb{R}_{\geqslant 0}$ satisfying the triangle inequality, requirements $r : V \to \mathbb{Z}_{\geqslant 1}$.
> **Output:** A tour that visits each $v \in V$ exactly $r(v)$ times.
> 1: Calculate an α-approximate solution $\mathsf{T}_{c,1}^\alpha$ for the single-visit metric TSP instance $(G, c, 1)$.
> 2: Calculate an optimal solution $\mathsf{TP}_{c,r-1}^\star$ for the transportation problem with prescriptions $r(v) - 1$ for $v \in V$.
> 3: **return** $T = \mathsf{T}_{c,1}^\alpha + \mathsf{TP}_{c,r-1}^\star$

Theorem 5. *The multigraph T returned by Algorithm 1 is a feasible solution to the metric* MANY-VISITS TSP *instance (G, c, r). The cost of the tour T is at most $(\alpha + 1) \cdot c(\mathsf{T}^\star_{c,r})$.*

Proof. The degree of each vertex $v \in V$ is 2 in $\mathsf{T}^\alpha_{c,1}$, and is $2 \cdot (r(v) - 1)$ in $\mathsf{TP}^\star_{c,r-1}$; hence the total degree of v in $T = \mathsf{T}^\alpha_{c,1} + \mathsf{TP}^\star_{c,r-1}$ is $2 \cdot r(v)$, as required. Since $\mathsf{T}^\alpha_{c,1}$ is connected, $T = \mathsf{T}^\alpha_{c,1} + \mathsf{TP}^\star_{c,r-1}$ is also connected, implying that it is a feasible solution to the problem.

The cost of the tour T constructed by Algorithm 1 is equal to $c(T) = c(\mathsf{T}^\alpha_{c,1}) + c(\mathsf{TP}^\star_{c,r-1})$. The cost of $\mathsf{T}^\alpha_{c,1}$ is at most $\alpha \cdot c(\mathsf{T}^\star_{c,1})$. Note that $c(\mathsf{T}^\star_{c,1}) \leqslant c(\mathsf{T}^\star_{c,r})$, as the cost function satisfies the triangle inequality. Again, by the triangle inequality, $c(\mathsf{TP}^\star_{c,r-1}) \leqslant c(\mathsf{TP}^\star_{c,r})$. Hence we get

$$
\begin{aligned}
c(T) &= c(\mathsf{T}^\alpha_{c,1}) + c(\mathsf{TP}^\star_{c,r-1}) \\
&\leqslant \alpha \cdot c(\mathsf{T}^\star_{c,1}) + c(\mathsf{TP}^\star_{c,r-1}) \\
&\leqslant \alpha \cdot c(\mathsf{T}^\star_{c,r}) + c(\mathsf{TP}^\star_{c,r}) \\
&\leqslant (\alpha + 1) \cdot c(\mathsf{T}^\star_{c,r}),
\end{aligned}
$$

proving the approximation guarantee stated in the theorem. $\qquad\square$

Christofides' algorithm [7] for the single-visit metric TSP provides an approximate solution with $\alpha = 1.5$; thus we get the following:

Corollary 6. *There is a polynomial-time algorithm that provides a 2.5-approximation for the metric* MANY-VISITS TSP.

Proof. The approximation ratio follows immediately; it remains to argue that the algorithm runs in polynomial time.

Finding an approximate solution for the single-visit TSP in Step 1 requires $\mathcal{O}(n^3)$ operations [7]. The transportation problem in Step 2 can be solved in $\mathcal{O}(n^3 \log r(V))$ operations using the Edmonds-Karp scaling method [10]. Finally, Step 3 takes $\mathcal{O}(n^2 \log r(V))$ operations, therefore the total time complexity of the algorithm is $\mathcal{O}(n^3 \log r(V))$. $\qquad\square$

For the metric MANY-VISITS ATSP, in Step 1 of Algorithm 1 we can apply the $\mathcal{O}(1)$-approximation for metric ATSP due to Svensson et al. [32]. This leads to the proof of Theorem 4.

4 Polyhedral background

In what follows, we make use of some basic notions and theorems of the theory of generalized polymatroids. For background, see for example the paper of Frank and Tardos [14] or Chapter 14 in the book by Frank [13].

Given a ground set S, a set function $b : 2^S \to \mathbb{Z}$ is *submodular* if

$$
b(X) + b(Y) \geqslant b(X \cap Y) + b(X \cup Y)
$$

holds for every pair of subsets $X, Y \subseteq S$. A set function $p : 2^S \to \mathbb{Z}$ is *supermodular* if $-p$ is submodular. As a generalization of matroid rank functions, Edmonds introduced the notion of polymatroids [9]. A set function b is a *polymatroid function* if $b(\varnothing) = 0$, b is non-decreasing, and b is submodular.

We define

$$
P(b) := \{ x \in \mathbb{R}^S_{\geqslant 0} : x(Y) \leqslant b(Y) \text{ for every } Y \subseteq S \} \ .
$$

The set of integral elements of $P(b)$ is called a *polymatroidal set*. Similarly, the *base polymatroid* $B(b)$ is defined by

$$
B(b) := \{ x \in \mathbb{R}^S : x(Y) \leqslant b(Y) \text{ for every } Y \subseteq S, \ x(S) = b(S) \} \ .
$$

Note that a base polymatroid is just a facet of the polymatroid $P(b)$. In both cases, b is called the *border function* of the polyhedron. Although non-negativity of x is not assumed in the definition of $B(b)$, this follows by the monotonicity of b and the definition of $B(b)$: $x(s) = x(S) - x(S - s) \geqslant b(S) - b(S - s) \geqslant 0$ holds for every $s \in S$. The set of integral elements of $B(b)$ is called a *base polymatroidal set*. Edmonds [9] showed that the vertices of a polymatroid or a base polymatroid are integral, thus $P(b)$ is the convex hull of the corresponding polymatroidal set, while $B(b)$ is the convex hull of the corresponding base polymatroidal set. For this reason, we will call the sets of integral elements of $P(b)$ and $B(b)$ simply a polymatroid and a base polymatroid.

Hassin [18] introduced polyhedra bounded simultaneously by a nonnegative, monotone non-decreasing submodular function b over a ground set S from above and by a nonnegative, monotone non-decreasing supermodular function p over S from below, satisfying the so-called *cross-inequality* linking the two functions:

$$b(X) - p(Y) \geqslant b(X - Y) - p(Y - X) \qquad \text{for every pair of subsets } X, Y \subseteq S \ .$$

We say that a pair (p, b) of set functions over the same ground set S is a *paramodular pair* if $p(\varnothing) = b(\varnothing) = 0$, p is supermodular, b is submodular, and the cross-inequality holds. The slightly more general concept of generalized polymatroids was introduced by Frank [12]. A *generalized polymatroid*, or *g-polymatroid* is a polyhedron of the form

$$Q(p, b) := \left\{ x \in \mathbb{R}^S : p(Y) \leqslant x(Y) \leqslant b(Y) \text{ for every } Y \subseteq S \right\} \ ,$$

where (p, b) is a paramodular pair. Here, (p, b) is called the *border pair* of the polyhedron. It is known (see e.g. [13]) that a g-polymatroid defined by an integral paramodular pair is a non-empty integral polyhedron.

A special g-polymatroid is a box $T(\ell, u) = \{x \in \mathbb{R}^S : \ell \leqslant x \leqslant u\}$ where $\ell : S \to \mathbb{Z} \cup \{-\infty\}$, $u : S \to \mathbb{Z} \cup \{\infty\}$ with $\ell \leqslant u$. Another illustrious example is base polymatroids. Indeed, given a polymatroid function b with finite $b(S)$, its *complementary set function* p is defined for $X \subseteq S$ by $p(X) := b(S) - b(S - X)$. It is not difficult to check that (p, b) is a paramodular pair and that $B(b) = Q(p, b)$.

The intersection Q' of a g-polymatroid $Q = Q(p, b)$ and a box $T = T(\ell, u)$ is non-empty if and only if $\ell(Y) \leqslant b(Y)$ and $p(Y) \leqslant u(Y)$ hold for every $Y \subseteq S$. When Q' is non-empty, its unique border pair (p', b') is given by

$$p'(Z) = \max\{p(Z') - u(Z' - Z) + \ell(Z - Z') : Z' \subseteq S\},$$

$$b'(Z) = \min\{b(Z') - \ell(Z' - Z) + u(Z - Z') : Z' \subseteq S\} \ .$$

Given a g-polymatroid $Q(p, b)$ and $Z \subset S$, by *deleting* $Z \subseteq S$ from $Q(p, b)$ we obtain a g-polymatroid $Q(p, b) \backslash Z$ defined on set $S - Z$ by the restrictions of p and b to $S - Z$, that is,

$$Q(p, b) \backslash Z := \{x \in \mathbb{R}^{S-Z} : p(Y) \leqslant x(Y) \leqslant b(Y) \text{ for every } Y \subseteq S - Z\} \ .$$

In other words, $Q(p, b) \backslash Z$ is the projection of $Q(p, b)$ to the coordinates in $S - Z$.

Extending the notion of contraction is not immediate. A set can be naturally identified with its characteristic vector, that is, contraction is basically an operation defined on $0 - 1$ vectors. In our proof, we will need a generalization of this to the integral elements of a g-polymatroid. However, such an element might have coordinates larger than one as well, hence finding the right definition is not straightforward. In the case of matroids, the most important property of contraction is the following: I is an independent of M/Z if and only if $F \cup I$ is independent in M for any maximal independent set F of Z.

With this property in mind, we define the g-polymatroid obtained by the contraction of an integral vector $z \in Q(p,b)$ to be the polymatroid $Q(p',b') := Q(p,b)/z$ on the same ground set S with the border functions

$$p'(X) := p(X) - z(X)$$
$$b'(X) := b(X) - z(X) \ .$$

Observe that p' is obtained as the difference of a supermodular and a modular function, implying that it is supermodular. Similarly, b' is submodular. Moreover, $p'(\varnothing) = b'(\varnothing) = 0$, and

$$b'(X) - p'(Y) = b(X) - z(X) - p(Y) + z(Y)$$
$$\geqslant b(X - Y) + p(Y - X) - z(X - Y) + z(Y - X)$$
$$= b'(X - Y) - p'(Y - X),$$

hence (p',b') is indeed a paramodular pair. The main reason for defining the contraction of an element $z \in Q(p,b)$ is shown by the following lemma.

Lemma 7. *Let $Q(p',b')$ be the polymatroid obtained by contracting $z \in Q(p,b)$. Then $x + z \in Q(p,b)$ for every $x \in Q(p',b')$.*

Proof. Let $x \in Q(p',b')$. By definition, this implies $p'(Y) \leqslant x(Y) \leqslant b'(Y)$ for $Y \subseteq S$. Thus $p(Y) = p'(Y) + z(Y) \leqslant x(Y) + z(Y) \leqslant b'(Y) + z(Y) = b(Y)$, concluding the proof. $\square$

Formally, the BOUNDED DEGREE G-POLYMATROID ELEMENT PROBLEM takes as input a g-polymatroid $Q(p,b)$ with a cost function $c : S \to \mathbb{R}$, and a hypergraph $H = (S, \mathcal{E})$ on the same ground set with lower and upper bounds $f, g : \mathcal{E} \to \mathbb{Z}_{\geqslant 0}$ and multiplicity vectors $m_\varepsilon : S \to \mathbb{Z}_{\geqslant 0}$ for $\varepsilon \in \mathcal{E}$ satisfying $m_\varepsilon(s) = 0$ for $s \in S - \varepsilon$. The objective is to find a minimum-cost element x of $Q(p,b)$ such that $f(\varepsilon) \leqslant \sum_{s \in \varepsilon} m_\varepsilon(s)x(s) \leqslant g(\varepsilon)$ for each $\varepsilon \in \mathcal{E}$.

5 Approximating the Bounded Degree g-polymatroid Element Problem with Multiplicities

The aim of this section is to prove Theorems 1 and 2. We start by formulating a linear programming relaxation for the BOUNDED DEGREE G-POLYMATROID ELEMENT PROBLEM:

$$
\begin{aligned}
& \text{minimize} && \sum_{s \in S} c(s)\, x(s) && \\
\text{(LP)} \quad & \text{subject to} && p(Z) \leqslant x(Z) \leqslant b(Z) && \forall Z \subseteq S \\
& && f(\varepsilon) \leqslant \sum_{s \in \varepsilon} m_\varepsilon(s)\, x(s) \leqslant g(\varepsilon) && \forall \varepsilon \in \mathcal{E}
\end{aligned}
$$

Although the program has an exponential number of constraints, it can be separated in polynomial time using submodular minimization [21, 26, 29]. Algorithm 2 generalizes the approach by Király et al. [23]. We iteratively solve the linear program, delete elements which get a zero value in the solution, update the solution values and perform a contraction on the polymatroid, or remove constraints arising from the hypergraph. There is a significant difference between the first round of the algorithm and the later ones. In the first round, the bounds on the coordinates solely depend on p and b, while in the subsequent rounds the whole problem is restricted to the unit cube. It is somewhat surprising that this restriction affects neither the solvability of the problem nor the additive error. Intuitively, the very first step of the algorithm fixes 'most part' of each coordinate, and the following steps are changing their value by at most 1.

Algorithm 2 Approximation algorithm for the BOUNDED DEGREE g-POLYMATROID ELEMENT PROBLEM WITH MULTIPLICITIES

Input: A g-polymatroid $Q(p,b)$ on ground set S, cost function $c : S \to \mathbb{R}$, a hypergraph $H = (S, \mathcal{E})$, lower bounds $f, g : \mathcal{E} \to \mathbb{Z}_{\geq 0}$, multiplicities $m_\varepsilon : S \to \mathbb{Z}_{\geq 0}$ for $\varepsilon \in \mathcal{E}$ satisfying $m_\varepsilon(s) = 0$ for $s \in S - \varepsilon$.

Output: $z \in Q(p,b)$ of cost at most OPT_{LP}, violating the hyperedge constraints by at most $2\Delta - 1$.

1: Initialize $z(s) \leftarrow 0$ for every $s \in S$.
2: **while** $S \neq \varnothing$ **do**
3:     Compute a basic optimal solution x for (LP).
    (Note: starting from the second iteration, $0 \leqslant x \leqslant 1$.)
    a: Delete any element s with $x(s) = 0$. Update each hyperedge $\varepsilon \leftarrow \varepsilon - s$ and $m_\varepsilon(s) \leftarrow 0$. Update the g-polymatroid $Q(p,b) \leftarrow Q(p,b) \backslash s$ by deletion.
    b: For all $s \in S$ update $z(s) \leftarrow z(s) + \lfloor x \rfloor(s)$.
    Apply polymatroid contraction $Q(p,b) \leftarrow Q(p,b)/\lfloor x \rfloor$, that is, redefine $p(Y) := p(Y) - \lfloor x \rfloor(Y)$ and $b(Y) := b(Y) - \lfloor x \rfloor(Y)$ for every $Y \subseteq S$.
    Update $f(\varepsilon) \leftarrow f(\varepsilon) - \sum_{s \in \varepsilon} m_\varepsilon(s) \lfloor x \rfloor(s)$ and $g(\varepsilon) \leftarrow g(\varepsilon) - \sum_{s \in \varepsilon} m_\varepsilon(s) \lfloor x \rfloor(s)$ for each $\varepsilon \in \mathcal{E}$.
    c: If $m_\varepsilon(\varepsilon) \leqslant 2\Delta - 1$, let $\mathcal{E} \leftarrow \mathcal{E} - \varepsilon$.
    d: **if** it is the first iteration **then**
    Take the intersection of $Q(p,b)$ and the unit cube $[0,1]^S$, that is, $p(Y) := \max\{p(Y') - |Y' - Y| : Y' \subseteq S\}$ and $b(Y) := \min\{b(Y') + |Y - Y'| : Y' \subseteq S\}$ for every $Y \subseteq S$.
4: **return** z

Proof of Theorem 1.

Correctness   First we show that if the algorithm terminates then the returned solution z satisfies the requirements of the theorem. In a single iteration, the g-polymatroid $Q(p,b)$ is updated to $(Q(p,b) \backslash D)/\lfloor x \rfloor$, where $D = \{s : x(s) = 0\}$ is the set of deleted elements. In the first iteration, the g-polymatroid thus obtained is further intersected with the unit cube. By Lemma 7, the vector $x - \lfloor x \rfloor$ restricted to $S - D$ remains a feasible solution for the modified linear program in the next iteration. Note that this vector is contained in the unit cube as its coordinates are between 0 and 1. This remains true when a lower degree constraint is removed in Step 3.c as well, therefore the cost of z plus the cost of an optimal LP solution does not increase throughout the procedure. Hence the cost of the output z is at most the cost of the initial LP solution, which is at most the optimum.

    By Lemma 7, the vector $x - \lfloor x \rfloor + z$ is contained in the original g-polymatroid, although it might violate some of the lower and upper bounds on the hyperdeges. We only remove the constraints corresponding to the lower and upper bounds for a hyperedge ε when $m_\varepsilon(\varepsilon) \leqslant 2\Delta - 1$. As the g-polymatroid is restricted to the unit cube after the first iteration, these constraints are violated by at most $2\Delta - 1$, as the total value of $\sum_{s \in \varepsilon} m_\varepsilon(s) z(s)$ can change by a value between 0 and $2\Delta - 1$ in the remaining iterations.

    It remains to show that the algorithm terminates successfully. The proof is based on similar arguments as in Király et al. [23, proof of Theorem 2].

Termination   Suppose, for sake of contradiction, that the algorithm does not terminate. Then there is some iteration after which none of the simplifications in Steps 3.a-3.c can be performed.

8

This implies that for the current basic LP solution x it holds $0 < x(s) < 1$ for each $s \in S$ and $m_\varepsilon(\varepsilon) \geqslant 2\Delta$ for each $\varepsilon \in \mathcal{E}$. We say that a set Y is *p-tight* (or *b-tight*) if $x(Y) = p(Y)$ (or $x(Y) = b(Y)$), and let $\mathcal{T}^p = \{Y \subseteq S : x(Y) = p(Y)\}$ and $\mathcal{T}^b = \{Y \subseteq S : x(Y) = b(Y)\}$ denote the collections of p-tight and b-tight sets with respect to solution x.

Let $\mathcal{L}$ be a maximal independent laminar system in $\mathcal{T}^p \cup \mathcal{T}^b$.

Claim 8. $\mathrm{span}\,(\{\chi_Z : Z \in \mathcal{L}\}) = \mathrm{span}\,(\{\chi_Z : Z \in \mathcal{T}^p \cup \mathcal{T}^b\})$

Proof of Claim 8. The proof uses an uncrossing argument. Let us suppose indirectly that there is a set R from $\mathcal{T}^p \cup \mathcal{T}^b$ for which $\chi_R \notin \mathrm{span}\,(\{\chi_Z : Z \in \mathcal{L}\})$. Choose this set R so that it is incomparable to as few sets of $\mathcal{L}$ as possible. Without loss of generality, we may assume that $R \in \mathcal{T}^p$. Now choose a set $T \in \mathcal{L}$ that is incomparable to R. Note that such a set necessarily exists as the laminar system is maximal. We distinguish two cases.
Case 1. $T \in \mathcal{T}^p$. Because of the supermodularity of p, we have

$$x(R) + x(T) = p(R) + p(T) \leqslant p(R \cup T) + p(R \cap T) \leqslant x(R \cup T) + x(R \cap T)$$
$$= x(R) + x(T) \ ,$$

hence equality holds throughout. That is, $R \cup T$ and $R \cap T$ are in $\mathcal{T}^p$ as well. In addition, since $\chi_R + \chi_T = \chi_{R \cup T} + \chi_{R \cap T}$ and χ_R is not in $\mathrm{span}\,(\{\chi_Z : Z \in \mathcal{L}\})$, either $\chi_{R \cup T}$ or $\chi_{R \cap T}$ is not contained in $\mathrm{span}\,(\{\chi_Z : Z \in \mathcal{L}\})$. However, both $R \cup T$ and $R \cap T$ are incomparable with fewer sets of $\mathcal{L}$ than R, which is a contradiction.
Case 2. $T \in \mathcal{T}^b$. Because of the cross-inequality, we have

$$x(T) - x(R) = b(T) - p(R) \geqslant b(T \backslash R) - p(R \backslash T) \geqslant x(T \backslash R) - x(R \backslash T)$$
$$= x(T) - x(R) \ ,$$

implying $T \backslash R \in \mathcal{T}^b$ and $R \backslash T \in \mathcal{T}^p$. Since $\chi_R + \chi_T = \chi_{R \backslash T} + \chi_{R \backslash T} + 2 \, \chi_{R \cup T}$ and χ_R is not in $\mathrm{span}\,(\{\chi_Z : Z \in \mathcal{L}\})$, one of the vectors $\chi_{R \backslash T}$, $\chi_{R \backslash T}$ and $\chi_{R \cup T}$ is not contained in $\mathrm{span}\,(\{\chi_Z : Z \in \mathcal{L}\})$. However, any of these three sets is incomparable with fewer sets of $\mathcal{L}$ than R, which is a contradiction.

The case when $R \in \mathcal{T}^b$ is analogous to the above. This completes the proof of the Claim. $\diamond$

We say that a hyperedge $\varepsilon \in \mathcal{E}$ is *tight* if $f(\varepsilon) = \sum_{s \in \varepsilon} m_\varepsilon(s) x(s)$ or $g(\varepsilon) = \sum_{s \in \varepsilon} m_\varepsilon(s) x(s)$. As x is a basic solution, there is a set $\mathcal{E}' \subseteq \mathcal{E}$ of tight hyperedges such that $\{m_\varepsilon : \varepsilon \in \mathcal{E}'\} \cup \{\chi_Z : Z \in \mathcal{L}\}$ are linearly independent vectors with $|\mathcal{E}'| + |\mathcal{L}| = |S|$.

We derive a contradiction using a token-counting argument. We assign 2Δ tokens to each element $s \in S$, accounting for a total of $2\Delta|S|$ tokens. The tokens are then redistributed in such a way that each hyperedge in $\mathcal{E}'$ and each set in $\mathcal{L}$ collects at least 2Δ tokens, while at least one extra token remains. This implies that $2\Delta|S| > 2\Delta|\mathcal{E}'| + 2\Delta|\mathcal{L}|$, leading to a contradiction.

We redistribute the tokens as follows. Each element s gives Δ tokens to the smallest member in $\mathcal{L}$ it is contained in, and $m_\varepsilon(s)$ token to each hyperedge $\varepsilon \in \mathcal{E}'$ it is contained in. As $\sum_{\varepsilon \in \mathcal{E}: s \in \varepsilon} m_\varepsilon(s) \leqslant \Delta$ holds for every element $s \in S$, thus we redistribute at most 2Δ tokens per element and so the redistribution step is valid. Now consider any set $U \in \mathcal{L}$. Recall that $\mathcal{L}^{\max}(U)$ consists of the maximal members of $\mathcal{L}$ lying inside U. Then $U - \bigcup_{W \in \mathcal{L}^{\max}(U)} W \neq \varnothing$, as otherwise $\chi_U = \sum_{W \in \mathcal{L}^{\max}(U)} \chi_W$, contradicting the independence of $\mathcal{L}$. For every set Z in $\mathcal{L}$, $x(Z)$ is an integer, meaning that $x(U - \bigcup_{W \in \mathcal{L}^{\max}(U)} W)$ is an integer. But also $0 < x(s) < 1$ for every $s \in S$, which means that $U - \bigcup_{W \in \mathcal{L}^{\max}(U)} W$ contains at least 2 elements. Therefore, each set U in $\mathcal{L}$ receives at least 2Δ tokens, as required. By assumption, $m_\varepsilon(\varepsilon) \geqslant 2\Delta$ for every hyperedge $\varepsilon \in \mathcal{E}'$, which means that each hyperedge in $\mathcal{E}'$ receives at least 2Δ tokens, as required.

If $\sum_{\varepsilon \in \mathcal{E}': s \in \varepsilon} m_\varepsilon(s) \leqslant \Delta$ holds for any $s \in S$ or $\mathcal{L}^{\max}(S)$ is not a partition of S, then an extra token exists. Otherwise, $\sum_{\varepsilon \in \mathcal{E}'} m_\varepsilon = \Delta \cdot \chi_S = \sum_{W \in \mathcal{L}^{\max}(S)}^{q} \chi_W$, contradicting the independence of $\{m_\varepsilon : \varepsilon \in \mathcal{E}'\} \cup \{\chi_Z : Z \in \mathcal{L}\}$.

Time complexity Let us now prove that the run time of the algorithm is polynomial in the input size. Solving an LP, as well as removing an element from a hyperedge in Step 3.c or removing a hyperedge in Step 3.a can be done in polynomial time. Now let us turn to the g-polymatroid contraction in Step 3.b and taking the intersection with the unit cube in Step 3.d. The function value is not recalculated for every subset $Y \subseteq S$, as there is an exponential number of such subsets. Instead, we calculate the value of the current functions p and b for a set Y only when it is needed during the ellipsoid method. We keep track of the vectors $\lfloor x \rfloor$ that arise during contraction steps (there is only a polynomial number of them), and every time a query for p or b happens, it takes into account every contraction and removal that occurred until that point.

Let us now bound the number of iterations. In every iteration at least one of Steps 3.a-3.c is executed. Clearly, Step 3.a can be repeated at most $|S|$ times, while Step 3.c can be repeated at most $|\mathcal{E}|$ times. Starting from the second iteration, we are working in the unit cube. That is, when Step 3.b adds the integer part of a variable $x(s)$ to $z(s)$ and reduces the problem, then the given variable will be 0 in the next iteration and so element s is deleted. This means that the total number of iterations of Step 3.b is at most $\mathcal{O}(|S|)$.

We therefore showed that the number of iterations, as well as the time complexity of each step taken by the algorithm can be bounded by the input size, meaning the algorithm runs in polynomial time. $\qquad\square$

Now we turn to the proof of the case when only lower or only upper bounds are given.

Proof of Theorem 2. The proof is similar to the proof of Theorem 1, the main difference appears in the the counting argument. When only lower bounds are present, the condition in Step 3.c changes: we delete a hyperedge ε if $f(\varepsilon) \leqslant \Delta - 1$. Suppose, for the sake of contradiction, that the algorithm does not terminate. Then there is an iteration after which none of the simplifications in Steps 3.a-3.c can be performed. This implies that in the current basic solution $0 < x(s) < 1$ holds for each $s \in S$ and $f(\varepsilon) \geqslant \Delta$ for each $\varepsilon \in \mathcal{E}$. We choose a subset $\mathcal{E}' \subseteq \mathcal{E}$ and a maximal independent laminar system $\mathcal{L}$ of tight sets the same way as in the proof of Theorem 1. Recall that $|\mathcal{E}'| + |\mathcal{L}| = |S|$.

Let $Z_1, \ldots, Z_k$ denote the members of the laminar system $\mathcal{L}$. As $\mathcal{L}$ is an independent system, $Z_i - \bigcup_{W \in \mathcal{L}^{\max}(Z_i)} W \neq \varnothing$. Since $x(s) < 1$ for all $s \in S$, $x(Z_i - \bigcup_{W \in \mathcal{L}^{\max}(Z_i)} W) < |Z_i - \bigcup_{W \in \mathcal{L}^{\max}(Z_i)} W|$. As we have integers on both sides of this inequality, we get

$$|Z_i - \bigcup_{W \in \mathcal{L}^{\max}(Z_i)} W| - x(Z_i - \bigcup_{W \in \mathcal{L}^{\max}(Z_i)} W) \geqslant 1 \quad \text{for all } i = 1, \ldots, k \ .$$

Moreover, $\sum_{s \in \varepsilon} m_\varepsilon(s) x(s) \geqslant f(\varepsilon) \geqslant \Delta$ for all hyperedges; therefore,

$$
\begin{aligned}
|\mathcal{E}'| + |\mathcal{L}| &\leqslant \sum_{\varepsilon \in \mathcal{E}'} \frac{\sum_{s \in \varepsilon} m_\varepsilon(s) x(s)}{\Delta} + \sum_{i=1}^{k} \left[|Z_i - \bigcup_{W \in \mathcal{L}^{\max}(Z_i)} W| - x(Z_i - \bigcup_{W \in \mathcal{L}^{\max}(Z_i)} W) \right] \\
&= \sum_{s \in S} \frac{x(s)}{\Delta} \sum_{\substack{\varepsilon \in \mathcal{E}' \\ s \in \varepsilon}} m_\varepsilon(s) + \sum_{W \in \mathcal{L}^{\max}(S)} |W| - \sum_{W \in \mathcal{L}^{\max}(S)} x(W) \\
&\leqslant |S| \ .
\end{aligned}
$$

10

In the last line, the first term is at most $x(S)$ since $\sum_{\varepsilon\in\mathcal{E}:s\in\varepsilon} m_\varepsilon(s) \leqslant \Delta$ holds for each element $s \in S$. From $x(S) - \sum_{W\in\mathcal{L}^{\max}(S)} x(W) \leqslant |S| - \sum_{W\in\mathcal{L}^{\max}(S)} |W|$ the upper bound of $|S|$ follows. As $|S| = |\mathcal{L}| + |\mathcal{E}'|$, we have equality throughout. This implies that $\sum_{\varepsilon\in\mathcal{E}'} m_\varepsilon = \Delta \cdot \chi_S = \Delta \cdot \sum_{W\in\mathcal{L}^{\max}(S)} \chi_W$, contradicting linear independence.

If only upper bounds are present, we remove a hyperedge ε in Step 3.c when $g(\varepsilon) + \Delta - 1 \geqslant m_\varepsilon(\varepsilon)$. Suppose, for the sake of contradiction, that the algorithm does not terminate. Then there is an iteration after which none of the simplifications in Steps 3.a-3.c can be performed. This implies that in the current basic solution $0 < x(s) < 1$ holds for each $s \in S$ and $m_\varepsilon(\varepsilon) - g(\varepsilon) \geqslant \Delta$ for each $\varepsilon \in \mathcal{E}$. Again, we choose a subset $\mathcal{E}' \subseteq \mathcal{E}$ and a maximal independent laminar system $\mathcal{L}$ of tight sets the same way as in the proof of Theorem 1.

Let $Z_1, \ldots, Z_k$ denote the members of the laminar system $\mathcal{L}$. As $\mathcal{L}$ is an independent system, $Z_i - \bigcup_{W\in\mathcal{L}^{\max}(Z_i)} W \neq \varnothing$ and so

$$x\left(Z_i - \bigcup_{W\in\mathcal{L}^{\max}(Z_i)} W\right) \geqslant 1 \ .$$

By $\sum_{s\in\varepsilon} m_\varepsilon(s)x(s) \leqslant g(\varepsilon)$, we get $\sum_{s\in\varepsilon} m_\varepsilon(s) - \sum_{s\in\varepsilon} m_\varepsilon(s)x(s) \geqslant m_\varepsilon(\varepsilon) - g(\varepsilon) \geqslant \Delta$. Therefore,

$$
\begin{aligned}
|\mathcal{E}'| + |\mathcal{L}| &\leqslant \sum_{\varepsilon\in\mathcal{E}'} \frac{\sum_{s\in\varepsilon} m_\varepsilon(s) - \sum_{s\in\varepsilon} m_\varepsilon(s)x(s)}{\Delta} + \sum_{i=1}^{k} x\left(Z_i - \bigcup_{W\in\mathcal{L}^{\max}(Z_i)} W\right) \\
&= \sum_{s\in S} \frac{1-x(s)}{\Delta} \sum_{\substack{\varepsilon\in\mathcal{E}' \\ s\in\varepsilon}} m_\varepsilon(s) + \sum_{W\in\mathcal{L}^{\max}(S)} x(W) \\
&\leqslant \sum_{s\in S} \frac{1-x(s)}{\Delta} \sum_{\substack{\varepsilon\in\mathcal{E}' \\ s\in\varepsilon}} m_\varepsilon(s) + x(S) \leqslant |S| \ .
\end{aligned}
$$

In the last line, the first term is at most $|S| - x(S)$ since $\sum_{\varepsilon\in\mathcal{E}:s\in\varepsilon} m_\varepsilon(s) \leqslant \Delta$ holds for every element $s \in S$. Therefore, the upper bound of $|S|$ follows. As $|S| = |\mathcal{L}| + |\mathcal{E}'|$, we have equality throughout. This implies that $\sum_{\varepsilon\in\mathcal{E}'} m_\varepsilon = \Delta \cdot \chi_S = \Delta \cdot \sum_{W\in\mathcal{L}^{\max}(S)} \chi_W$, contradicting linear independence. $\square$

We have seen in Sect. 4 that base polymatroids are special cases of g-polymatroids. This implies that the results of Theorem 2 immediately apply to polymatroids. Let us first formally define the problem.

In the LOWER BOUNDED DEGREE POLYMATROID BASIS PROBLEM WITH MULTIPLICITIES, we are given a base polymatroid $B(b) = (S, b)$ with a cost function $c : S \to \mathbb{R}$, and a hypergraph $H = (S, \mathcal{E})$ on the same ground set. The input contains lower bounds $f : \mathcal{E} \to \mathbb{Z}_{\geqslant 0}$ and multiplicity vectors $m_\varepsilon : \varepsilon \to \mathbb{Z}_{\geqslant 1}$ for every hyperedge $\varepsilon \in \mathcal{E}$. The objective is to find a minimum-cost element $x \in B(b)$ such that $f(\varepsilon) \leqslant \sum_{s\in\varepsilon} m_\varepsilon(s)x(s)$ holds for each $\varepsilon \in \mathcal{E}$.

Corollary 9. *There is an algorithm for the* LOWER BOUNDED DEGREE POLYMATROID BASIS PROBLEM WITH MULTIPLICITIES *that runs in polynomial time and returns an element x of $B(b)$ of cost at most the optimum value such that $f(\varepsilon) - \Delta + 1 \leqslant \sum_{s\in\varepsilon} m_\varepsilon(s)x(s)$ for each $\varepsilon \in \mathcal{E}$.*

6   A 1.5-Approximation for the Metric Many-Visits TSP

In this section we design a polynomial-time 1.5-approximation for the METRIC MANY-VISITS TSP. Our approach is along similar lines as Christofides' algorithm [7] for the metric single-visit TSP. It constructs a solution in three steps: (i) it computes a minimum cost spanning tree that ensures the connectivity of the solution, then (ii) it adds a minimum cost matching on the set of vertices of odd degree in order to obtain an Eulerian subgraph, and finally (iii) it forms a Hamiltonian circuit from an Eulerian circuit by shortcutting repeated vertices.

In our setting of many-visits, we make use of the following formulation of the metric MANY-VISITS TSP: Given a complete undirected graph G with non-negative cost function $c : E \to \mathbb{Z}_{\geq 0}$ and requirements $r : V \to \mathbb{Z}_{\geq 1}$, find a vector $x \in \mathbb{Z}_{\geq 0}^E$ minimizing $c^T x$ such that $d_x(v) = 2r(v)$ for every $v \in V$, and $\mathrm{supp}(x)$ is connected. From now on we use $\hat{r} = r(V) - |V| + 1$.

The high-level idea of the algorithm is the following. We first show that the set of integral vectors $\{x \in \mathbb{Z}_{\geq 0}^E : x(E) = r(V),\ \mathrm{supp}(x)\ \text{is connected}\}$ form the integral points of a base polymatroid. We apply Corollary 9 to this base polymatroid to obtain a vector $x \in \mathbb{Z}_{\geq 0}^E$ with $c^T x$ no more than the optimum, such that $d_x(v) \geq 2r(v) - 1$ for $v \in V$. Then we add a minimum-cost matching on the set of vertices of odd $d_x(v)$-value. Finally, by shortcutting vertices with degree higher than prescribed, we obtain a tour that satisfies the requirements on the number of visits at every vertex.

Lemma 10. *Let b denote the following function defined on edge sets $F \subseteq E$:*

$$(1) \qquad\qquad b(F) = \begin{cases} |V(F)| - \mathrm{comp}(F) + \hat{r} & \text{if } F \neq \varnothing, \\ 0 & \text{otherwise.} \end{cases}$$

Then b is a polymatroid function.

Proof. By definition, $b(\varnothing) = 0$ and b is monotone increasing. It remains to show that b is submodular. Let $X, Y \subseteq E$. The submodular inequality clearly holds if one of X and Y is empty. If none of X and Y is empty then the submodular inequality follows from the fact that $|V(F)| - \mathrm{comp}(F)$ is the rank function of the graphical matroid. $\qquad\square$

Consider the base polymatroid $B(b)$ determined by the border function defined in (1). Let us define the set $B = \{x \in \mathbb{Z}_{\geq 0}^E : x(E) = r(V),\ \mathrm{supp}(x)\ \text{is connected}\}$.

Lemma 11. $B = B(b) \cap \mathbb{Z}_{\geq 0}^E$.

Proof. Take an integral element $x \in B(b)$ and let $C \subseteq E$ be an arbitrary cut between V_1 and V_2 for some partition $V_1 \uplus V_2$ of V. Then

$$
\begin{aligned}
x(C) &= x(E) - (x(E(V_1) \cup E(V_2))) \\
&\geq |V| - 1 + \hat{r} - (|V_1| + |V_2| - \mathrm{comp}(E(V_1) \cup E(V_2)) + \hat{r}) \\
&\geq 1,
\end{aligned}
$$

thus $\mathrm{supp}(x)$ is connected. As $x(E) = |V| - 1 + \hat{r} = r(V)$, we obtain $x \in B$, showing that $B(b) \subseteq B$.

To see the other direction, take an element $x \in B$. As $\mathrm{supp}(x)$ is connected, $x(E - F) \geq \mathrm{comp}(F) + |V| - |V(F)| - 1$ for every $F \subseteq E$. That is,

$$
\begin{aligned}
x(F) &= x(E) - x(E - F) \\
&\leq r(V) - (|V - V(F)| + \mathrm{comp}(F) - 1) \\
&= |V(F)| - \mathrm{comp}(F) + \hat{r},
\end{aligned}
$$

thus $x(F) \leq b(F)$. As $x(E) = r(V) = |V| - 1 + \hat{r}$, we obtain $x \in B(b)$, showing $B \subseteq B(b)$. $\quad\square$

Algorithm 3 A 1.5-approximation algorithm for the metric MANY-VISITS TSP

Input: A complete undirected graph G, costs $c : E \to \mathbb{R}_{\geqslant 0}$ satisfying the triangle inequality, requirements $r : V \to \mathbb{Z}_{\geqslant 1}$.

Output: A tour that visits each $v \in V$ exactly $r(v)$ times.
1: Construct the polymatroid $B(b) = (S, b)$, where $S := E$ and b is defined as in Equation (1).
2: Construct a hypergraph $H = (S, \mathcal{E})$ with $\mathcal{E} = \{\delta(v) : v \in V\}$ and
   ○ for every $\varepsilon \in \mathcal{E}$ and $s \in \varepsilon$, set $m_\varepsilon(s) = 2$ if s is a self-loop and $m_\varepsilon(s) = 1$ otherwise,
   ○ for every $\varepsilon \in \mathcal{E}$, set $f(\varepsilon) = 2 \cdot r(v)$, where $\varepsilon = \delta(v)$.
3: Run Algorithm 2 with $B(b), c, H, f$ and the m_ε's as input. Let $z \in B(b)$ denote the output.
4: Calculate a minimum-cost matching M with respect to c on the vertices of V with odd $d_z(v)$ values.
5: Determine a tour $T = \{C, \mu_C\}_{C \in \mathcal{C}}$ from z and χ_M.
6: Do shortcuts in T and obtain a solution T', such that T' visits every city v exactly $r(v)$ times (that is, $d_T(v) = r(v)$ for every vertex $v \in V$).
7: **return** T'.

Our algorithm is presented as Algorithm 3. First, we construct a polymatroid B and a hypergraph H, such that their common ground set S consists of the edges of the graph G in our MANY-VISITS TSP instance. The border function b of the polymatroid is defined in Equation (1).

For each vertex v of G, there is a hyperedge ε in the hypergraph that contains all edges of G incident to v, including the self-loop at v. We set the multiplicities of an element $s \in S$ to 2 if it corresponds to a self-loop in G, and to 1 otherwise. The motivation is that a self-loop contributes the the degree of a vertex by two, while a regular edge contributes to the degree of each of its endpoints by one. Note that an element $s \in S$ is contained in exactly one hyperedge if it corresponds to a self-loop, and it is contained in exactly two hyperedges otherwise; therefore the total contribution of each edge adds up to two.

Now we are ready to prove Theorem 3.

Proof of Theorem 3. First, let us show that Algorithm 3 provides a feasible solution for the given instance (G, c, r) of the metric MANY-VISITS TSP.

By Lemma 11, the solution z provided by Algorithm 2 in Step 2 is such that $c^T z \leqslant c(\mathsf{T}^\star_{c,r})$, $z(E) = r(V)$ and $\text{supp}(z)$ is connected. Note that in our case $\Delta = \max_{s \in S}\{\sum_{\varepsilon \in \mathcal{E} : s \in \varepsilon} m_\varepsilon(s)\}$ = 2. By Theorem 2, $f(\varepsilon) - 1 \leqslant \sum_{s \in \varepsilon} m_\varepsilon(s) z(s)$ for each $\varepsilon \in \mathcal{E}$, and this inequality translates to $2 \cdot r(v) - 1 \leqslant d_z(v)$ for every $v \in V$. That is, z corresponds to a multigraph of cost at most $c(\mathsf{T}^\star_{c,r})$ violating the degree prescriptions from below by at most one. Note that this means the total violation from above is at most $|V| - 1$. In Step 3 we calculate a matching M that provides one extra degree to each odd-degree vertex. That is, in the union of the multigraph defined by z and M, every vertex v has an even degree.

Constructing a tour and shortcutting   In Step 4, we construct a *compact representation* of a tour from the vector z and matching M, and we denote it by T. We use the algorithm described in Grigoriev et al. [17], which takes the edge multiplicities as input, and outputs a collection $\mathcal{C}$ of pairs (C, μ_C). Here C is a simple closed walk, and μ_C is the corresponding integer denoting the number of copies of the walk C in T. From the pairs (C, μ_C) it is possible to construct an implicit order of the vertices the following way.

Let us construct an auxiliary multigraph A on the vertex set V by taking the edges of each cycle C exactly once. Note that parallel edges are allowed in A if an edge appears in multiple cycles C. Due to the construction, every vertex has an even degree in A, which means that there

exist an Eulerian circuit in A. Moreover, there are $\mathcal{O}(n^2)$ distinct cycles [17], hence, the total number of edges in A is $\mathcal{O}(n^3)$. Consequently, using Hierholzer's algorithm, we can compute an Eulerian circuit η in A in $\mathcal{O}(n^3)$ time [11, 19]. The circuit η covers the edges of each cycle C once. Now an implicit order of the vertices in the MANY-VISITS TSP tour T is the following. Traverse the vertices of the Eulerian circuit η in order. Every time a vertex u appears the first time, traverse all cycles C that contain the vertex $\mu_C - 1$ times. Denote this circuit by η'. It is easy to see that the sequence η' is a sequence of vertices that uses the edges of each cycle C exactly μ_C times, meaning this is a feasible sequence of the vertices in the tour T. Moreover, the order itself takes polynomial space, as it is enough to store indices of $\mathcal{O}(n^3)$ vertices and $\mathcal{O}(n^2)$ cycles.

Now let us consider the set W of vertices w that have more visits than $r(w)$ in the tour T. Denote the surplus of visits of a vertex $w \in W$ by $\gamma(w) := d_T(w) - 2 \cdot r(w)$. In Step 6, we remove the last $\gamma(w)$ occurrences of every vertex $w \in W$ from T, by doing shortcuts: if an occurrence of w is preceded by u and superseded by v in T, replace the edges uw and wv by uv in the sequence. This can be done by traversing the compact representation of η' backwards, and removing the vertex w from the last $\gamma(w)$ cycles $C^{(w)}_{r(w)-\gamma(w)+1}, \ldots, C^{(w)}_{r(w)}$. As $\sum_w \gamma(w)$ can be bounded by $\mathcal{O}(n)$, this operation makes $\mathcal{O}(n)$ new cycles, keeping the space required by the new sequence of vertices and cycles polynomial. Moreover, since the edge costs are metric, making shortcuts the way described above cannot increase the total cost of the edges in T. Finally, using a similar argument as in the algorithm of Christofides, the shortcutting does not make the tour disconnected. The resulting graph is therefore a tour T' that visits every vertex v exactly $r(v)$ times, that is, a feasible solution for the instance (G, c, r).

Cost and complexity   The cost of the edges in z is at most $c(\mathsf{T}^\star_{c,r})$, and as the cost function satisfies the triangle inequality, the cost of the matching M found in Step 3 is at most $c(\mathsf{T}^\star_{c,1})/2$. Moreover, taking shortcuts at vertices does not increase the cost of the solution, hence the cost of the output is at most $c^T(z + \chi_M) \leq c(\mathsf{T}^\star_{c,r}) + c(\mathsf{T}^\star_{c,1})/2 = 1.5 \cdot c(\mathsf{T}^\star_{c,r})$.

Now we turn to the complexity analysis. All edge multiplicities during the algorithm are stored as integer numbers in binary, therefore the space needed of any variable representing multiplicities of edges can be bounded by $\mathcal{O}(n^2 \log \sum r(v))$. Steps 1-2 can be performed in time that is polynomial in the input size. The function b is defined in Lemma 10 and can be computed efficiently. Therefore, according to Corollary 9, the algorithm in Step 3 also runs in polynomial time. Step 4 can also be done in polynomial time [17], and the number of closed walks can be bounded by $\mathcal{O}(n^2)$. Moreover, the total surplus of degrees in T is at most $n - 1$, therefore the number of shortcutting operations is also bounded by n. This completes the proof. $\qquad\square$

It is worth considering what Algorithm 3 does when applied to the single-visit TSP, that is, when $r(v) = 1$ for each $v \in V$. The output of Algorithm 2 in Step 3 is a connected multigraph with $r(V) = n$ edges. Note that the guarantee that each vertex v has degree at least $2 \cdot r(v) - 1 = 1$ does not add anything as this already follows from connectivity. Such a graph is basically the union of a spanning tree and a single edge (where the edge might be also part of the spanning tree, that is, in the solution having multiplicity 2); we call such a graph a *1-tree*. The rest of the algorithm mimics Christofides' algorithm: a minimum cost matching is added on the set of vertices of odd degree to get an Eulerian graph, and then a Hamiltonian circuit is formed by shortcutting repeated vertices in an Eulerian circuit. That is, applying our algorithm to a single-visit TSP instance, it is almost identical to that of Christofides, except that instead of a spanning tree we start with a 1-tree. However, the 1-tree we start with is not necessarily a cheapest one among all possible choices; we only know that its cost is at most the cost of the optimal single-visit TSP tour.

7 Discussion

In this work we developed an approximation algorithm for the minimum-cost degree bounded g-polymatroid element problem with multiplicities. The approximation algorithm yields a solution of cost at most the optimum, which violates the lower bounds only by a constant factor depending on the weighted maximum element frequency Δ. We then demonstrated the usefulness of our result by developing a polynomial-time 1.5-approximation algorithm for the metric many-visits traveling salesman problem. This way, we match the famous Christofides-Serdyukov bound for the single-visit TSP.

References

[1] N. Bansal, R. Khandekar, J. Könemann, V. Nagarajan, and B. Peis. On generalizations of network design problems with degree bounds. *Math. Prog.*, 141(1):479–506, 2013.

[2] N. Bansal, R. Khandekar, and V. Nagarajan. Additive guarantees for degree-bounded directed network design. *SIAM J. Comput.*, 39(4):1413–1431, 2009.

[3] A. Berger, L. Kozma, M. Mnich, and R. Vincze. A time- and space-optimal algorithm for the many-visits TSP. In *Proc. SODA 2019*, pages 1770–1782, 2019.

[4] N. Brauner, Y. Crama, A. Grigoriev, and J. van de Klundert. A framework for the complexity of high-multiplicity scheduling problems. *J. Combinatorial Optim.*, 9(3):313–323, 2005.

[5] K. Chaudhuri, S. Rao, S. Riesenfeld, and K. Talwar. A push–relabel approximation algorithm for approximating the minimum-degree MST problem and its generalization to matroids. *Theoret. Comput. Sci.*, 410(44):4489–4503, 2009.

[6] K. Chaudhuri, S. Rao, S. Riesenfeld, and K. Talwar. What would Edmonds do? Augmenting paths and witnesses for degree-bounded MSTs. *Algorithmica*, 55(1):157–189, 2009.

[7] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Technical Report 388, Carnegie Mellon University, 1976.

[8] S. S. Cosmadakis and C. H. Papadimitriou. The traveling salesman problem with many visits to few cities. *SIAM J. Comput.*, 13(1):99–108, 1984.

[9] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and their Applications (Proc. Calgary Internat. Conf., Calgary, Alta., 1969)*, pages 69–87. Gordon and Breach, New York, 1970.

[10] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. In *Combinatorial Structures and their Applications (Proc. Calgary Internat. Conf., Calgary, Alta., 1969)*, pages 93–96. Gordon and Breach, New York, 1970.

[11] H. Fleischner. *Eulerian graphs and related topics. Part 1. Vol. 2*, volume 50 of *Annals of Discrete Mathematics*. North-Holland Publishing Co., Amsterdam, 1991.

[12] A. Frank. Generalized polymatroids. In *Finite and infinite sets*, pages 285–294. Elsevier, 1984.

[13] A. Frank. Connections in combinatorial optimization. *Discrete Appl. Math.*, 160(12):1875, 2012.

[14] A. Frank and É. Tardos. Generalized polymatroids and submodular flows. *Math. Prog.*, 42(1-3):489–563, 1988.

[15] M. Fürer and B. Raghavachari. Approximating the minimum-degree Steiner tree to within one of optimal. *J. Algorithms*, 17(3):409–423, 1994.

[16] M. X. Goemans. Minimum bounded degree spanning trees. In *Proc. FOCS 2006*, pages 273–282, 2006.

[17] A. Grigoriev and J. van de Klundert. On the high multiplicity traveling salesman problem. *Discrete Optim.*, 3(1):50–62, 2006.

[18] R. Hassin. Minimum cost flow with set-constraints. *Networks*, 12(1):1–21, 1982.

[19] C. Hierholzer and C. Wiener. Ueber die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren. *Math. Ann.*, 6(1):30–32, 1873.

[20] D. S. Hochbaum and R. Shamir. Strongly polynomial algorithms for the high multiplicity scheduling problem. *Oper. Res.*, 39(4):648–653, 1991.

[21] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM*, 48(4):761–777, 2001.

[22] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[23] T. Király, L. C. Lau, and M. Singh. Degree bounded matroids and submodular flows. *Combinatorica*, 32(6):703–720, 2012.

[24] J. Könemann and R. Ravi. A matter of degree: Improved approximation algorithms for degree-bounded minimum spanning trees. *SIAM J. Comput.*, 31(6):1783–1793, 2002.

[25] J. Könemann and R. Ravi. Primal-dual meets local search: approximating MST's with nonuniform degree bounds. In *Proc. STOC 2003*, pages 389–395, 2003.

[26] S. T. McCormick. Submodular function minimization. *Handbooks in operations research and management science*, 12:321–391, 2005.

[27] H. N. Psaraftis. A dynamic programming approach for sequencing groups of identical jobs. *Oper. Res.*, 28(6):1347–1359, 1980.

[28] M. Rothkopf. Letter to the editor–the traveling salesman problem: On the reduction of certain large problems to smaller ones. *Oper. Res.*, 14(3):532–533, 1966.

[29] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *J. Combinatorial Theory, Ser. B*, 80(2):346–355, 2000.

[30] A. Serdyukov. On some extremal walks in graphs. *Upravlyaemye systemy*, 17:76–79, 1978.

[31] M. Singh and L. C. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *Proc. STOC 2007*, pages 661–670, 2007.

[32] O. Svensson, J. Tarnawski, and L. A. Végh. A constant-factor approximation algorithm for the asymmetric traveling salesman problem. In *Proc. STOC 2018*, pages 204–213. 2018.

[33] J. A. A. van der Veen and S. Zhang. Low-complexity algorithms for sequencing jobs with a fixed number of job-classes. *Comput. Oper. Res.*, 23(11):1059–1067, 1996.

[34] R. Zenklusen. Matroidal degree-bounded minimum spanning trees. In *Proc. SODA 2012*, pages 1512–1521, 2012.