Additive Sparsification of CSPs

EDEN PELLEG, University of Oxford, United Kingdom
STANISLAV ŽIVNÝ, University of Oxford, United Kingdom

Multiplicative cut sparsifiers, introduced by Benczúr and Karger [STOC'96], have proved extremely influential and found various applications. Precise characterisations were established for sparsifiability of graphs with other 2-variable predicates on Boolean domains by Filtser and Krauthgamer [SIDMA'17] and non-Boolean domains by Butti and Živný [SIDMA'20].

Bansal, Svensson and Trevisan [FOCS'19] introduced a weaker notion of sparsification termed "additive sparsification", which does not require weights on the edges of the graph. In particular, Bansal et al. designed algorithms for additive sparsifiers for cuts in graphs and hypergraphs.

As our main result, we establish that *all* Boolean Constraint Satisfaction Problems (CSPs) admit an additive sparsifier; that is, for every Boolean predicate $P : \{0,1\}^k \to \{0,1\}$ of a fixed arity k, we show that CSP(P) admits an additive sparsifier. Under our newly introduced notion of all-but-one sparsification for non-Boolean predicates, we show that CSP(P) admits an additive sparsifier for *any* predicate $P : D^k \to \{0,1\}$ of a fixed arity k on an arbitrary finite domain D.

CCS Concepts: • **Theory of computation → Sparsification and spanners**.

Additional Key Words and Phrases: constraint satisfaction, sparsification

ACM Reference Format:
Eden Pelleg and Stanislav Živný. 2023. Additive Sparsification of CSPs. *ACM Trans. Algor.* 1, 1, Article 1 (January 2023), 19 pages.

1 INTRODUCTION

Graph sparsification is the problem of, given a graph $G = (V, E)$ with quadratically many (in $|V|$) edges, finding a sparse subgraph $G_\varepsilon = (V, E_\varepsilon \subseteq E)$ such that important properties of G are preserved in G_ε. Sparse in this context usually means with sub-quadratically many edges, though in this work we require (and can achieve) linearly many edges.

One of the most studied properties of preservation is the size of cuts. If $G = (V, E, w)$ is an undirected weighted graph with $w : E \to \mathbb{R}_{>0}$, given some $S \subseteq V$, the cut of S in G is

$$\mathrm{Cut}_G(S) = \sum_{\substack{\{u,v\}\in E \\ |\{u,v\}\cap S|=1}} w(\{u,v\}),$$

the sum of weights of all edges connecting S and $S^c = V \setminus S$. In an influential paper, Benczúr and Karger [11] introduced cut sparsification with a multiplicative error. In particular, [11] showed that for any graph $G = (V, E, w)$ and any error parameter $0 < \varepsilon < 1$, there exists a sparse subgraph

Authors' addresses: Eden Pelleg, University of Oxford, Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom; Stanislav Živný, standa.zivny@cs.ox.ac.uk, University of Oxford, Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2023 Association for Computing Machinery.
1549-6325/2023/1-ART1 $15.00
https://doi.org/

$G_\varepsilon = (V, E_\varepsilon \subseteq E, w')$ with $O(n(\log n)\varepsilon^{-2})$ edges (and new weights w' on the edges in E_ε), such that for every $S \subseteq V$ we have

$$\mathrm{Cut}_{G_\varepsilon}(S) \in (1 \pm \varepsilon)\mathrm{Cut}_G(S).$$

This was later improved by Batson, Spielman and Srivastava [10] to a subgraph with $O(n\varepsilon^{-2})$ many edges. Andoni, Chen, Krauthgamer, Qin, Woodruff and Zhang showed that the dependency on ε is optimal [4].

The ideas from cut sparsification paved the way to various generalisations, including streaming [1], sketching [4], cuts in hypergraphs [23, 27], spectral sparsification [21, 31–34] and the consideration of other predicates besides cuts [20]. In this work, we focus on the latter.

The cut sparsification result in [10] was explored for other Boolean binary predicates by Filtser and Krauthgamer [20], following a suggestion to do so by Kogan and Krauthgamer in [23]. Filtser and Krauthgamer found [20] a necessary and sufficient condition on the predicate for the graph to be sparsifiable (in the sense of [10]). In particular, [20] showed that not all Boolean binary predicates are sparsifiable. Later, Butti and Živný [14] generalised the result from [20] to arbitrary finite domain binary predicates.

We remark that [14, 20] use the terminology of *constraint satisfaction problems* (CSPs) with a fixed predicate P. This is is equivalent to a (hyper)graph G with a fixed predicate. Indeed, the vertices of G correspond to the variables of the CSP and the (hyper)edges of G correspond to the constraints of the CSP. If the fixed predicate P is not symmetric, the (hyper)edges of G are directed. We will mostly talk about sparsification of (hyper)graphs with a fixed predicate but this is equivalent to the CSP view.

Recently, while trying to eliminate the requirement for the introduction of new weights for the sparse subgraph, Bansal, Svensson and Trevisan [7] have come up with a new sparsification notion with an additive error term. They have shown (cf. Theorem 3 in Section 2) that under their notion any undirected unweighted hypergraph has a sparse subhypergraph which preserves all cuts up to some additive term.

Motivation. The relatively recent notion of additive sparsification has not yet been explored to the same extent as the notion of multiplicative sparsification has been. We believe that this notion has a lot of potential for applications as the sparsifiers are not weighted, unlike multiplicative sparsifiers, and the main restriction of multiplicative sparsifiers in applications appears to be the number of distinct weights required in sparsifiers. For some graphs (such as the "barbell graph" – two disjoint cliques joined by a single edge), any nontrivial multiplicative sparsifier requires edges of different weights. In any case, the authors find the notion of additive sparsification interesting in its own right, independently of applications. We refer the reader to [7] for further details and a discussion.

The goal of our work is to understand how the notion of additive sparsification developed in [7] for cuts behaves on (hyper)graphs with other predicates (beyond cuts), deriving inspiration from the generalisations of cuts to other predicates in the multiplicative setting established in [14, 20]. In particular, already Boolean binary predicates include interesting predicates such as the *uncut edges* (using the predicate $P(x, y) = 1$ iff $x = y$), *covered edges* (using the predicate $P(x, y) = 1$ iff $x = 1$ or $y = 1$), or *directed cut edges* (using the predicate $P(x, y) = 1$ iff $x = 0$ and $y = 1$). While such graph problems are well-known and extensively studied, it is not clear whether one should expect them to be sparsifiable or not. For instance, as mentioned before, not all (even Boolean binary) predicates are sparsifiable multiplicatively [20]. Are there some predicates that are not additively sparsifiable?

1.1 Contributions

Boolean predicates. Our main result, Theorem 10 in Section 3, shows that *all* hypergraphs with *constant* uniformity k, directed or undirected, admit additive sparsification with respect to all Boolean predicates $P : \{0,1\}^k \longrightarrow \{0,1\}$; the number of hyperedges of the sparsifier with error $\varepsilon > 0$ is $O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right)$, where the $O(\cdot)$ hides a factor that depends on k. This result has three ingredients. First, we observe that the result in [7] also holds true for directed hypergraphs. Second, we use a reduction via the k-partite k-fold covers of hypergraphs to the already solved case of Boolean Cut. Finally, we use linear algebra to prove the correctness of the reduction. While the reduction via the k-partite k-fold cover was used in previous works on multiplicative sparsification [14, 20], the subsequent non-trivial linear-algebraic analysis (Proposition 14) is novel and constitutes our main technical contribution, as well as our result that, unlike in the multiplicative setting, all Boolean predicates can be (additively) sparsified. We also show that our results immediately apply to the more general setting where different hyperedges are associated with different predicates (cf. Remark 11). This corresponds to CSPs with a fixed constraint language (of a finite size) rather than just a single predicate.

Non-Boolean predicates. We introduce a notion of sparsification that generalises the Boolean case to predicates on non-Boolean domains, i.e. a notion capturing predicates of the form $P : D^k \longrightarrow \{0,1\}$, where D is an arbitrary fixed finite set with $|D| \geq 2$. We call this type of sparsification "all-but-one" sparsification since the additive error term includes the maximum volume of $|D|-1$ (out of $|D|$) parts, where the volume of a subset is the sum of the degrees in the subset. (The precise definition can be found in Section 4.) By building on the techniques used to establish our main result, we show that all hypergraphs (again, directed or undirected) admit additive all-but-one sparsification with respect to all predicates. This is stated as Theorem 21 in Section 4. We also show, in Section 5, that our notion of all-but-one sparsification is, in some sense, optimal.

Comparison to previous work. As mentioned above, our sparsifiability result is obtained by a reduction via the k-partite k-fold cover to the cut case established in [7]. A reduction via the k-partite k-fold cover was also used (for $k = 2$) in previous work on multiplicative sparsification [14, 20]. In particular, the correctness of the reduction for Boolean binary predicates in [20] is done via an ad hoc case analysis for 11 concrete predicates. In the generalisation to binary predicates on arbitrary finite domains in [14], the correctness is proved via a combinatorial property of bipartite graphs without a certain 4-vertex graph[1] as a subgraph and a reduction to cuts with more than two parts.

In our case, we use the same black-box reduction via the k-partite k-fold cover. Thus the reduction itself is pretty straightforward, although the analysis is not. In fact, we find it surprising and unexpected that the k-partite k-fold cover works in the additive setting. Our key contribution is the proof of its correctness. A few simple reductions get us to the most technically involved case, in which k is even and the k-ary predicate satisfies $P(1,\ldots,1) = 0$. Additive sparsifiability of such predicates is established in Proposition 14. Unlike in the multiplicative setting, it is not clear how to do this in a straightforward way similar to [14, 20]. Instead, we associate with a given predicate P a vector v_P in an appropriate vector space, identify special vectors that can be shown additively sparsifiable directly, show that linear combinations preserve sparsifiability, and argue that v_P can be generated by the special vectors. The latter is the most technical part of the proof. While there are several natural ideas how to achieve this in a seemingly simpler way (such as arguing that the special vectors form a basis), we have not managed to produce a simpler or shorter proof.

The result in [7] also works for non-constant k. We emphasise that we deal with constant k, which is standard in the CSP literature in that the predicate (or a set of predicates) is fixed and not

[1]A bipartite graph on four vertices with each part of size two and precisely one edge between the two parts.

part of the input. For constant k, the representation of predicates is irrelevant (cf. Remark 18). Thus we do not keep track of (and have not tried to optimise) the precise dependency of the reduction on the predicate arity k (or the domain size $q = |D|$).

Related work. The already mentioned spectral sparsification [33] is a stronger notion than cut sparsification as it requires that not only cuts but also the Laplacian spectrum of a given graph should be (approximately) preserved [7, 21, 31, 32, 34].

Our focus in this article is on *edge sparsifiers* (of cuts and generalisations via local predicates). There are also vertex sparsifiers, in which one reduces the number of vertices. Vertex sparsifiers have been studied for cut sparsification (between special vertices called terminals) [15, 22, 25, 26] as well as for spectral sparsification [24].

Sparsification in general is about finding a sparse sub(hyper)graph while preserving important properties of interest. In addition to cut sparsifiers, another well studied concept is that of *spanners*. A spanner of a graph is a (sparse) subgraph that approximately preserves distances of shortest paths. Spanners have been studied in great detail both in the multiplicative [3, 5, 6, 9, 17, 28, 30] and additive [2, 8, 12, 16, 18, 35] setting. Emulators are a generalisation of spanners in which the sparse graph is not required to be a subgraph of the original graph. We refer the reader to a nice recent survey of Elkin and Neimain for more details [19].

2 PRELIMINARIES

For an integer k, we denote by $[k]$ the set $\{0, 1, \dots, k-1\}$. All graphs and hypergraphs[2] in this paper are unweighted.

For an assignment $a : V \longrightarrow S$ from the set of vertices of a (hyper)graph to some set S containing 0, we denote by $\mathrm{Z}_a = \{v \in V : a(v) = 0\}$ the set of vertices mapped to 0.

If $0 \le i \le r^k - 1$ is an integer, we denote by $\mathrm{rep}_{r,k}(i)$ the representation of i in base r as a vector in $\mathbb{R}^k$, where the first coordinate stands for the most significant digit, and the last coordinate for the least significant digit. For the special case $r = 2$, we use the notation $\mathrm{bin}_k(i)$ for the binary representation of i.

We denote by $v[j]$ the j-th coordinate of the vector v, counting from 0.

For an integer $0 \le i \le 2^k - 1$, we use $\mathrm{zeros}_k(i) = \{\ell \in [k] : \mathrm{bin}_k(i)[\ell] = 0\}$; for example $\mathrm{zeros}_6(52) = \{2, 4, 5\}$, since $\mathrm{bin}_6(52) = (1, 1, 0, 1, 0, 0)$.

We now define the value of an assignment on a hypergraph with a fixed predicate.

Definition 1. Let $G = (V, E)$ be a directed k-uniform hypergraph and let $P : D^k \longrightarrow \{0, 1\}$ be a k-ary predicate on a finite set D. Given an assignment $a : V \longrightarrow D$ of G, the *value* of a is defined by $\mathrm{Val}_{G,P}(a) = \sum_{(v_1,\dots,v_k)\in E} P(a(v_1), \dots, a(v_k))$. If G is undirected and P is order invariant,[3] we define $\mathrm{Val}_{G,P}(a) = \sum_{\{v_1,\dots,v_k\}\in E} P(a(v_1), \dots, a(v_k))$.[4]

The notion of additive sparsification was first introduced in [7] for cuts in graphs and hypergraphs. In order to define it, we will need the Cut : $\{0, 1\}^k \longrightarrow \{0, 1\}$ predicate defined by $\mathrm{Cut}(b_1, \dots, b_k) = 1 \iff \exists i, j, b_i \ne b_j$. Given a hypergraph $G = (V, E)$ and a set $U \subseteq V$, we denote by $\mathrm{vol}_G(U)$ the *volume* of U, defined as the sum of the degrees in G of all vertices in U.

Definition 2. Let $G = (V, E)$ be an undirected k-uniform hypergraph, and denote $|V| = n$. We say that G *admits additive cut sparsification* with error ε using $O(f(n, \varepsilon))$ hyperedges if there exists a subhypergraph $G_\varepsilon = (V, E_\varepsilon \subseteq E)$ with $|E_\varepsilon| = O(f(n, \varepsilon))$, called an *additive sparsifier* of G, such that

[2]We use the standard definition of hypergraphs, in which every hyperedge is an ordered tuple of vertices.
[3]$P(b_1, \dots, b_k) = P(b_{\sigma(1)}, \dots, b_{\sigma(k)})$ for all $b_1, \dots, b_k \in D$ and every permutation σ on the set $\{1, \dots, k\}$.
[4]The terms are well defined since P is order invariant.

for every assignment $a : V \longrightarrow \{0,1\}$ we have

$$\left|\frac{|E|}{|E_\varepsilon|}\mathrm{Val}_{G_\varepsilon,\mathrm{Cut}}(a) - \mathrm{Val}_{G,\mathrm{Cut}}(a)\right| \le \varepsilon(d_G|\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a)), \tag{1}$$

where d_G is the average degree of G.

Note that (1) can also be written as

$$\frac{|E|}{|E_\varepsilon|}\mathrm{Val}_{G_\varepsilon,\mathrm{Cut}}(a) \in \mathrm{Val}_{G,\mathrm{Cut}}(a) \pm \varepsilon(d_G|\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a)),$$

which explains the use of the term "additive" for the error.

Bansal, Svensson and Trevisan [7] showed the following sparsification result:

Theorem 3 (Additive Cut Sparsification [7, Theorem 1.3]). *Let $G = (V, E)$ be an undirected n-vertex k-uniform hypergraph, and $\varepsilon > 0$. Then G admits additive cut sparsification with error ε using $O\left(\frac{n}{k}\varepsilon^{-2}\log(\frac{k}{\varepsilon})\right)$ hyperedges.*

Remark 4. We call a predicate P *symmetric* if it is order invariant (as in Definition 1). Since Theorem 3 deals with only *undirected* hypergraphs, it is not clear how to generalise it to non-symmetric predicates directly, since the value of such predicates on undirected hypergraphs is not defined. Therefore, our course of action will be first to prove it for the case of directed hypergraphs, and then generalise it to other predicates on directed hypergraphs. In fact, by doing this we also prove the result for undirected hypergraphs with symmetric predicates, since hyperedges can be given arbitrary directions without changing the average degree of G, or the volume in G, or the value of the predicate in any assignment.

Remark 5. Throughout this paper we only discuss the existence of sparsifiers and do not mention the time complexity to find them. However, the (implicit) time complexity results from [7] apply in our more general setting as well since the sparsifiers we find are in fact the same sparsifiers for all predicates, including cuts (cf. Remark 17).

An important tool we use to prove our results is the k-partite k-fold cover of a hypergraph. This construction is a well known one, and has been used for multiplicative sparsification (for $k = 2$) in [20] and [14].

Definition 6. Let $G = (V, E)$ be a directed k-uniform hypergraph. The *k-partite k-fold cover* of G is the hypergraph $\gamma(G) = (V^\gamma, E^\gamma)$ where

$$V^\gamma = \{v^{(0)}, v^{(1)}, \ldots, v^{(k-1)} : v \in V\},$$

$$E^\gamma = \{(v_1^{(0)}, v_2^{(1)}, \ldots, v_k^{(k-1)}) : (v_1, \ldots, v_k) \in E\}.$$

If G is undirected we define the cover in the same way except

$$E^\gamma = \{\{v_{\sigma(1)}^{(0)}, v_{\sigma(2)}^{(1)}, \ldots, v_{\sigma(k)}^{(k-1)}\} : \{v_1, \ldots, v_k\} \in E\},\ \sigma \text{ a permutation on } \{1, 2, \ldots, k\}\}$$

so for each hyperedge in G we get $k!$ hyperedges in $\gamma(G)$ in this case.

If $k = 2$ then $\gamma(G)$ corresponds to the well-known *bipartite double cover* of G [13].

3 SPARSIFICATION OF BOOLEAN PREDICATES

As mentioned in Section 1, we begin by observing that Theorem 3 also works for directed hypergraphs. (We emphasise that we treat k as a constant, cf. Remark 18.)

We will need a notation for the undirected equivalent of a directed hypergraph.

Definition 7. Given a directed k-uniform hypergraph $G = (V, E)$, the *undirected equivalent* of G is $\Lambda(G) = (V, \overline{E})$ where $\overline{E} = \{\{v_1, \dots, v_k\} : (v_1, \dots, v_k) \in E\}$.

In other words, $\Lambda(G)$ is obtained by "forgetting" the directions of the hyperedges of G (and ignoring duplicates if they exist).

Proposition 8. *Let $G = (V, E)$ be a directed n-vertex k-uniform hypergraph, and $\varepsilon > 0$. Then G admits additive cut sparsification with error ε using $O\left(n\varepsilon^{-2} \log \frac{1}{\varepsilon}\right)$ hyperedges.*

Proof. Let $\varepsilon > 0$, and let $\gamma(G) = (V^\gamma, E^\gamma)$ be the k-partite k-fold cover of G. Let $\Lambda(\gamma(G))$ be the undirected equivalent of $\gamma(G)$. Let $\Lambda(\gamma(G))_\varepsilon = (V^\gamma_\varepsilon, \overline{E^\gamma_\varepsilon})$ be a subhypergraph of $\Lambda(\gamma(G))$ promised by Theorem 3. By the construction of the k-partite k-fold cover, there are no two directed hyperedges over the same set of vertices, and so there is a 1-1 correspondence between the hyperedges of G and the hyperedges of $\Lambda(\gamma(G))$. Hence we have a subhypergraph $G_\varepsilon = (V, E_\varepsilon)$ of G such that $\Lambda(\gamma(G_\varepsilon)) = \Lambda(\gamma(G))_\varepsilon$ (by taking the hyperedges corresponding to the ones of $\Lambda(\gamma(G))_\varepsilon$). We also have $|\overline{E^\gamma}| = |E|$ and $|\overline{E^\gamma_\varepsilon}| = |E_\varepsilon|$.

Let $a : V \longrightarrow \{0, 1\}$. Define $a' : V^\gamma \longrightarrow \{0, 1\}$ by $a'(v^{(i)}) = a(v)$. We have

$$\mathrm{Val}_{G,\mathrm{Cut}}(a) = \mathrm{Val}_{\Lambda(\gamma(G)),\mathrm{Cut}}(a'), \tag{2}$$

which is true for any hypergraph, and in particular for G_ε:

$$\mathrm{Val}_{G_\varepsilon,\mathrm{Cut}}(a) = \mathrm{Val}_{\Lambda(\gamma(G_\varepsilon)),\mathrm{Cut}}(a'). \tag{3}$$

Applying Theorem 3 to $\Lambda(\gamma(G))$ and a' gives us

$$\begin{aligned}\left|\frac{|E|}{|E_\varepsilon|}\mathrm{Val}_{G_\varepsilon,\mathrm{Cut}}(a) - \mathrm{Val}_{G,\mathrm{Cut}}(a)\right| &= \left|\frac{|\overline{E^\gamma}|}{|\overline{E^\gamma_\varepsilon}|}\mathrm{Val}_{\Lambda(\gamma(G_\varepsilon)),\mathrm{Cut}}(a') - \mathrm{Val}_{\Lambda(\gamma(G)),\mathrm{Cut}}(a')\right| \\ &\le \varepsilon(d_{\Lambda(\gamma(G))}\,|\mathrm{Z}_{a'}| + \mathrm{vol}_{\Lambda(\gamma(G))}(\mathrm{Z}_{a'})) \\ &= \varepsilon(d_{\gamma(G)} \cdot k|\mathrm{Z}_a| + \mathrm{vol}_{\gamma(G)}(\mathrm{Z}_{a'})) \\ &= \varepsilon(d_G|\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a)),\end{aligned}$$

where the first line is due to (2) and (3), the second line is by Theorem 3, and the last two lines are by properties of the k-partite k-fold cover. Moreover

$$|E_\varepsilon| = |\overline{E^\gamma_\varepsilon}| = O\left(\frac{kn}{k}\varepsilon^{-2}\log\frac{k}{\varepsilon}\right) = O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right),$$

as required. □

From now on, whenever we say a "hypergraph", we mean a "directed hypergraph" with n vertices. By Remark 4, the results also apply to undirected hypergraphs (whenever it makes sense, i.e. if the associated predicate is symmetric). We also omit the word additive when discussing sparsification. The following notion of sparsification is a natural generalisation of cut sparsification (Definition 2) to arbitrary predicates.

Definition 9. Let P be a k-ary Boolean predicate and $G = (V, E)$ a k-uniform hypergraph. We say that G *admits P-sparsification* with error ε using $O(f(n, \varepsilon))$ hyperedges if there exists a sub-hypergraph $G_\varepsilon = (V, E_\varepsilon \subseteq E)$ with $|E_\varepsilon| = O(f(n, \varepsilon))$, called a *$P$-sparsifier* of G, such that for every assignment $a : V \longrightarrow \{0, 1\}$ we have

$$\left| \frac{|E|}{|E_\varepsilon|} \mathrm{Val}_{G_\varepsilon, P}(a) - \mathrm{Val}_{G,P}(a) \right| \leq \varepsilon(d_G |\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a)), \tag{4}$$

where d_G is the average degree of G.

The following theorem is our main result, extending Proposition 8 to *all* k-ary predicates with Boolean domains.

Theorem 10 (**Main**). *For every k-uniform hypergraph G (k is a constant), every k-ary Boolean predicate $P : \{0, 1\}^k \longrightarrow \{0, 1\}$, and every $\varepsilon > 0$, G admits P-sparsification with error ε using $O\left(n\varepsilon^{-2} \log \frac{1}{\varepsilon}\right)$ hyperedges.*

Theorem 10 can be informally restated as "*every k-uniform hypergraph is sparsifiable with respect to all k-ary Boolean predicates*" or "*for every Boolean predicate P of constant arity,* CSP(P) *is sparsifiable*".

Remark 11. It is possible to consider an even more general case where each hyperedge in G has its own predicate. In this case, we can apply Theorem 10 to each of the hypergraphs obtained by taking only hyperedges corresponding to a specific predicate, and so get a sparsifier for each such predicate. Taking the union of all their hyperedges, we get a new hypergraph G_ε, which is a sparsifier of the original hypergraph. Indeed, it has $O\left(n\varepsilon^{-2} \log \frac{1}{\varepsilon}\right)$ hyperedges since it is the union of a constant number of hypergraphs. (The number of predicates $P : \{0, 1\}^k \longrightarrow \{0, 1\}$ is constant, since k is constant.) It also satisfies (4) for any given assignment up to some constant factor, since all the sparsifiers it is composed of do. This constant factor can be eliminated by choosing $\varepsilon_0 = \frac{\varepsilon}{m}$ for an appropriate m that depends only on k.

The main work in the proof of Theorem 10 is for even values of k; a simple reduction (Proposition 16) then reduces the case of k odd to the even case.

In order to prove Theorem 10 for even k, we use the k-partite k-fold cover of G and apply Proposition 8 to various assignments of it. For a k-ary Boolean predicate $P : \{0, 1\}^k \longrightarrow \{0, 1\}$, we consider the vector $v_P \in \mathbb{R}^{2^k}$, defined by $v_P[i] = P(\mathrm{bin}_k(i))$. For instance, for the Cut predicate on a 3-uniform hypergraph, we have $v_{\mathrm{Cut}} = (0, 1, 1, 1, 1, 1, 1, 0)$.

For a given hypergraph G and an assignment a, we consider the vector $v_{G,a} \in \mathbb{R}^{2^k}$ defined by $v_{G,a}[i] = |\{(v_1, \ldots, v_k) \in E : (a(v_1), \ldots, a(v_k)) = \mathrm{bin}_k(i)\}|$. In other words, each coordinate of $v_{G,a}$ counts the hyperedges in G whose vertices are assigned some specific set of values by a.

Example 12. Given the graph $G = (V, E)$ in Figure 1 (so $k = 2$) and the assignment $a : V \longrightarrow \{0, 1\}$ defined as $a(v_1) = a(v_2) = a(v_3) = 0$ and $a(v_4) = a(v_5) = a(v_6) = a(v_7) = 1$, we have $v_{G,a} = (2, 3, 1, 5)$, since there are two edges with assignment $(0, 0)$, namely (v_1, v_2) and (v_2, v_3), three edges with assignment $(0, 1)$, namely (v_1, v_4), (v_2, v_6), and (v_2, v_7), etc.

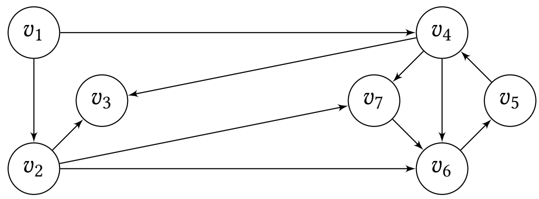

Fig. 1. Graph from Example 12.

Under these notations, we get $\mathrm{Val}_{G,P}(a) = \langle v_P, v_{G,a}\rangle$, where $\langle\cdot,\cdot\rangle$ is the standard inner product in $\mathbb{R}^{2^k}$. We begin by proving the following useful lemma.

Lemma 13. *Let $G = (V, E)$ be a k-uniform hypergraph, $P_1, \ldots, P_m$ be k-ary Boolean predicates (m is a constant). Suppose that for every $\varepsilon > 0$ and $1 \le i \le m$, G admits P_i-sparsification with error ε using $O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right)$ hyperedges, and that the same subhypergraph $G_\varepsilon = (V, E_\varepsilon \subseteq E)$ is a P_i-sparsifier for all P_i. Suppose that P is some k-ary Boolean predicate for which we have $v_P = \sum_{i=1}^m \lambda_i v_{P_i}$ for some constants $\lambda_1, \ldots, \lambda_m \in \mathbb{R}$. Under these conditions, G admits P-sparsification with error ε using $O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right)$ hyperedges.*

Proof. Let $\varepsilon > 0$ and denote $\varepsilon_i = \frac{\varepsilon}{m|\lambda_i|}$ (if $\lambda_i = 0$ take $\varepsilon_i = 1$ instead) and $\varepsilon_0 = \min\{\varepsilon_1, \ldots, \varepsilon_m\}$. Let $G_{\varepsilon_0} = (V, E_{\varepsilon_0})$ be the common witness subhypergraph for ε_0 promised by the assumption. We know that every P_i satisfies

$$\left|\frac{|E|}{|E_{\varepsilon_0}|}\mathrm{Val}_{G_{\varepsilon_0},P_i}(a) - \mathrm{Val}_{G,P_i}(a)\right| \le \varepsilon_0(d_G|\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a)) \tag{5}$$

for every assignment $a : V \longrightarrow \{0,1\}$. We also have

$$\mathrm{Val}_{G,P}(a) = \langle v_P, v_{G,a}\rangle = \left\langle \sum_{i=1}^m \lambda_i v_{P_i}, v_{G,a}\right\rangle = \sum_{i=1}^m \lambda_i\langle v_{P_i}, v_{G,a}\rangle = \sum_{i=1}^m \lambda_i \mathrm{Val}_{G,P_i}(a),$$

and similarly

$$\mathrm{Val}_{G_{\varepsilon_0},P}(a) = \sum_{i=1}^m \lambda_i \mathrm{Val}_{G_{\varepsilon_0},P_i}(a).$$

Therefore, for every assignment a we get

$$\begin{aligned}\left|\frac{|E|}{|E_{\varepsilon_0}|}\mathrm{Val}_{G_{\varepsilon_0},P}(a) - \mathrm{Val}_{G,P}(a)\right| &= \left|\frac{|E|}{|E_{\varepsilon_0}|}\sum_{i=1}^m \lambda_i\mathrm{Val}_{G_{\varepsilon_0},P_i}(a) - \sum_{i=1}^m \lambda_i\mathrm{Val}_{G,P_i}(a)\right| \\ &\le \sum_{i=1}^m |\lambda_i|\left|\frac{|E|}{|E_{\varepsilon_0}|}\mathrm{Val}_{G_{\varepsilon_0},P_i}(a) - \mathrm{Val}_{G,P_i}(a)\right| \\ &\le \sum_{i=1}^m |\lambda_i|\varepsilon_0(d_G|\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a)) \\ &\le \varepsilon(d_G|\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a)),\end{aligned}$$

where the second line is due to the triangle inequality, the third is due to (5) and the fourth is by the definition of ε_0.

Furthermore, since m and all λ_i are constants,

$$|E_{\varepsilon_0}| = O\left(n\varepsilon_0^{-2}\log\frac{1}{\varepsilon_0}\right) = O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right),$$

and so G_{ε_0} is a witness for the P-sparsification of G. □

The core of the proof of Theorem 10 is in the next proposition, which establishes the result for Boolean predicates on even uniformity hypergraphs, with a small restriction.

Proposition 14. *Let k be an even number and G be a k-uniform hypergraph. Let P be a k-ary Boolean predicate with $P(1,1,\ldots,1)=0$. Then for every $\varepsilon>0$, G admits P-sparsification with error ε using $O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right)$ hyperedges.*

Proof. Let $\varepsilon>0$. We consider $\gamma(G)$, the k-partite k-fold cover of G. Let $\gamma(G)_\varepsilon$ be a subhypergraph of $\gamma(G)$ promised by Proposition 8, and $G_\varepsilon=(V,E_\varepsilon)$ the corresponding subhypergraph of G, i.e. the subhypergraph which satisfies $\gamma(G_\varepsilon)=\gamma(G)_\varepsilon$ (by taking the hyperedges corresponding to the ones of $\gamma(G)_\varepsilon$).

Let $a:V\longrightarrow\{0,1\}$. For every subset $T\subseteq[k]$, we look at the assignment $a_T:V^\gamma\longrightarrow\{0,1\}$ defined by $a_T(v^{(i)})=0$ if $i\in T$ and $a(v)=0$, and $a_T(v^{(i)})=1$ otherwise. We therefore have

$$\left|\frac{|E^\gamma|}{|E^\gamma_\varepsilon|}\mathrm{Val}_{\gamma(G)_\varepsilon,\mathrm{Cut}}(a_T)-\mathrm{Val}_{\gamma(G),\mathrm{Cut}}(a_T)\right|\le\varepsilon(d_{\gamma(G)}|\mathrm{Z}_{a_T}|+\mathrm{vol}_{\gamma(G)}(\mathrm{Z}_{a_T})).\tag{6}$$

Define the vector $u_T\in\mathbb{R}^{2^k}$ as follows:

$$u_T[j]=\begin{cases}1 & T\cap\mathrm{zeros}(j)\neq\emptyset,[k]\\ 0 & otherwise\end{cases}.$$

In other words, the vector u_T is 1 in index j if and only if there exists an index $i\in T$ in which the binary representation of j has a zero, with the exception of $u_{[k]}[0]=0$. Denote by P_T the predicate corresponding to u_T, that is $P_T(\mathrm{bin}_k(j))=1\iff u_T[j]=1$. Observe that

$$\mathrm{Val}_{\gamma(G),\mathrm{Cut}}(a_T)=\mathrm{Val}_{G,P_T}(a),$$

since they both count exactly hyperedges $(v_1,\ldots,v_k)$ which have some vertex v_i with $a(v_i)=0$ with $i\in T$, but if $T=[k]$ then they do not count hyperedges which have $a(v_i)=0$ for all $i=1,\ldots,k$ (see example in Figure 2). The same is true for any hypergraph, and in particular for G_ε, that is

$$\mathrm{Val}_{\gamma(G_\varepsilon),\mathrm{Cut}}(a_T)=\mathrm{Val}_{G_\varepsilon,P_T}(a).$$

Putting these results in (6), we get

$$\begin{aligned}\left|\frac{|E|}{|E_\varepsilon|}\mathrm{Val}_{G_\varepsilon,P_T}(a)-\mathrm{Val}_{G,P_T}(a)\right|&\le\varepsilon(d_{\gamma(G)}|\mathrm{Z}_{a_T}|+\mathrm{vol}_{\gamma(G)}(\mathrm{Z}_{a_T}))\\&\le\varepsilon(d_G|\mathrm{Z}_a|+\mathrm{vol}_G(\mathrm{Z}_a)),\end{aligned}$$

so G admits P_T sparsification with error ε using $O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right)$ hyperedges for every $T\subseteq[k]$, and for every ε the sparsification is witnessed by the same subhypergraph G_ε. (Notice that Proposition 8, when applied to $\gamma(G)$ which has kn vertices, gives us a subhypergraph with $O\left(kn\varepsilon^{-2}\log\frac{1}{\varepsilon}\right)$ hyperedges, and recall that k is a constant.)

Our next goal is to show that the vector v_P is a linear combination of the vectors u_T for all $T\in[k]$. To show that, we show that every vector e_r in the standard basis of $\mathbb{R}^{2^k}$, with $r\neq 2^k-1$, is a linear combination of these vectors. This is sufficient since the last coordinate of v_P is 0 by the assumption. First we need to order the various sets T. We order them in the following decreasing lexicographic order $T_0,T_1,\ldots,T_{2^k-1}$, where $T_j=\mathrm{zeros}(j)$, so $T_0=[k],T_1=[k]\setminus\{k-1\},T_2=[k]\setminus\{k-2\},T_3=[k]\setminus\{k-1,k-2\},T_4=[k]\setminus\{k-3\}$ and so on, until $T_{2^k-1}=\emptyset$.

Let e_r be a vector in the standard basis of $\mathbb{R}^{2^k}$. We introduce the following coefficients for $0\le m\le 2^k-1$:

$$\lambda_{r,m}=\frac{1}{2}(-1)^{\mathrm{Ham}(r\oplus m)+(1-\mathbb{1}_{r\&m})},$$

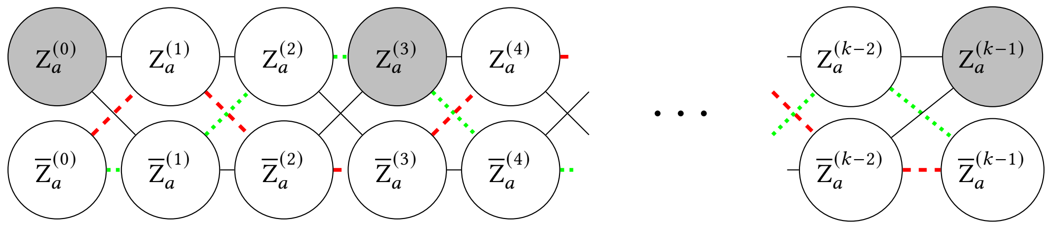

Fig. 2. An example of a representation of an assignment on $\gamma(G)$. $Z_a^{(i)}$ consists of all vertices in $V^{(i)}$ which are a copy of a vertex $v \in V$ with $a(v) = 0$, and $\overline{Z}_a^{(i)}$ consists of the rest of $V^{(i)}$. Each hyperedge has a unique path from left to right (but a path might belong to multiple hyperedges), choosing one of $Z_a^{(i)}, \overline{Z}_a^{(i)}$ for each i. Each such path is also in 1-1 correspondence with a coordinate in u_T. In this example $T = \{0, 3, k-1\}$ and the shaded sets represent $a_T^{-1}(0)$. By green dotted lines we indicated a path corresponding to a hyperedge counted in $\mathrm{Val}_{\gamma(G),\mathrm{Cut}}(a_T)$, and by red dashed lines we indicated a path which does not. The green dotted path corresponds to a value of 1 in the coordinate of u_T with binary representation $(1, 1, 0, 0, 1, \ldots, 0, 1)$, and the red dashed path to a value 0 in the coordinate with binary representation $(1, 0, 1, 1, 0, , \ldots, 1, 1)$. Note that if $T = [k]$ then any hyperedge corresponding to a path only on $Z_a^{(i)}$ is not counted.

where $\oplus, \&$ are the Xor and And binary functions respectively,[5] Ham is the Hamming weight function, and $\mathbb{1}_d$ returns 1 if $d \neq 0$ and 0 if $d = 0$. Denote

$$f_1(m) = \mathrm{Ham}(r \oplus m) \quad , \quad f_2(m) = (1 - \mathbb{1}_{r\&m}).$$

We shall prove that

$$e_r = \sum_{m=0}^{2^k-1} \lambda_{r,m} u_{T_m}. \tag{7}$$

We start with a claim.

Claim: The sum of all coefficients is 0; i.e., $\sum_{m=0}^{2^k-1} \lambda_{r,m} = 0$.

Proof of the claim. Let $b_1 b_2 \ldots b_k$ be the binary representation of r. Since $r < 2^k - 1$, there exists some $1 \leq i \leq r$ for which $b_i = 0$. We can partition the coefficients into pairs, such that $\lambda_{r,m_1}, \lambda_{r,m_2}$ is a pair if and only if m_1, m_2 differ in the i-th coordinate only. This is clearly a partition. For each pair, f_1 gives m_1, m_2 different parity values, and f_2 gives them the same value (since $b_i = 0$), so $\lambda_{r,m_1}, \lambda_{r,m_2}$ have opposite signs, so their sum is zero. This is true for every pair, so the overall sum is zero, and the claim is proved. *(End of the proof of the claim.)*

We prove (7) coordinate-wise. First we look at the coordinate r. Consider the set W of all vectors u_{T_m} for which the coordinate r is 0. If we show that the sum of the corresponding coefficients of the vectors in W is -1, using the claim we will deduce the result in this case. We distinguish 2 cases:

Case (I): $r = 0$. By the definition of u_T, in this case the set W contains two vectors, $u_{[k]}$ and $u_\emptyset$. The corresponding coefficients are $\lambda_{r,0} = -\frac{1}{2}$ and $\lambda_{r,2^k-1} = -\frac{1}{2}$ (since k is even), which sum up to -1.

Case (II): $r > 0$. As in the proof of the claim, let $b_1 b_2 \ldots b_k$ be the binary representation of r, and choose a coordinate $1 \leq i \leq k$ for which $b_i = 1$. Partition the vectors in W into pairs where $u_{T_{m_1}}, u_{T_{m_2}}$ is a pair if and only if m_1, m_2 differ in the i-th coordinate only. This is clearly a partition of all vectors, and by the definition of u_T, each such pair is either contained in W or disjoint from W so this is indeed a partition of W. (Note that $u_{T_{m_1}}[r]$ is determined by $T_{m_1} \cap \mathrm{zeros}(r)$ which is

[5]The Xor of two integers is defined as the bitwise Boolean Xor of their binary representations, where the Boolean Xor of two bits is their sum modulo 2. The And of two integers is defined the same way with the Boolean And function which is defined as $\mathrm{And}(i, j) = 1 \iff i = j = 1$.

in fact $\text{zeros}(m_1) \cap \text{zeros}(r)$, and the same for m_2. Since m_1, m_2 differ in the i-th coordinate only, and r is not zero in this coordinate, this coordinate contributes nothing to the intersections, and so both these intersections are empty or non-empty together. The intersection never equals $[k]$ since $r > 0$.) For every such pair in W, if it does not contain the negation of $\text{bin}(r)$, then there is some other index $j \neq i$ in which r, m_1, m_2 are all 1. (This is because in all other coordinates m_1, m_2 are equal, and since they are not the negation of r, there is some coordinate $j \neq i$ in which they are equal to the j-th coordinate of r. These coordinates cannot be all 0, since this would imply $u_{T_{m_1}}, u_{T_{m_2}} \notin W$.) This implies that f_2 gives m_1, m_2 the same value, and clearly f_1 gives them different parity values, so $\lambda_{r,m_1} + \lambda_{r,m_2} = 0$. However, for the pair which contains the negation of r (this pair is clearly in W), suppose without loss of generality the negation is m_1. Then f_2 gives m_1, m_2 the values $1, 0$ respectively, and f_1 gives m_1 an even value and m_2 an odd value (since k is even), and so $\lambda_{r,m_1} = \lambda_{r,m_2} = -\frac{1}{2}$, and the overall sum is -1. This finishes the proof of (7) in the coordinate r.

Now let $r' \neq r$ be some other coordinate, and let $c_1 c_2 \dots c_k$ be its binary representation. First, if $r' = 2^k - 1$ then for all m we have $u_{T_m}[r'] = 0$ by definition, so the linear combination of this coordinate is 0. So suppose $r' < 2^k - 1$. As before let W be the set of all vectors u_{T_m} for which the coordinate r' is 0. We show that the sum of the corresponding coefficients is zero, and again deduce the result using the claim. Now, there exists some index i for which $b_i \neq c_i$. Again we have two cases:

Case (1): $b_i = 0, c_i = 1$. Partition the vectors in W into pairs where $u_{T_{m_1}}, u_{T_{m_2}}$ is a pair if and only if m_1, m_2 differ in the i-th coordinate only. This is clearly a partition of all the vectors, and by the definition of u_T, each such pair is either contained in W or disjoint from W, so this is indeed a partition of W. For every such pair in W, f_1 gives m_1, m_2 different parity values, and f_2 gives them the same value (since $b_i = 0$), so $\lambda_{r,m_1}, \lambda_{r,m_2}$ have opposite signs, so their sum is zero. This is true for every pair in W, so the overall sum is zero.

Case (2) $b_i = 1, c_i = 0$. Here we consider two sub-cases:

Case (2a): $r' = 0$. The only vectors in W in this case are $u_{[k]}$ and $u_\emptyset$. The corresponding coefficients are $\lambda_{r,0} = \frac{1}{2}(-1)^{\text{Ham}(r)+1}$ and $\lambda_{r,2^k-1} = \frac{1}{2}(-1)^{\text{Ham}(\neg r)}$, where $\neg r$ denotes the negation of the binary representation of r. Since k is even, we know that $r, \neg r$ have the same parity, and so the sum of the two coefficients is 0.

Case (2b): $r' \neq 0$. Choose some j for which $c_j = 1$. Partition the vectors in W into pairs where $u_{T_{m_1}}, u_{T_{m_2}}$ is a pair if and only if m_1, m_2 differ in the j-th coordinate only. The argument for this being a partition of W is similar to the argument in Case (1). For each pair in W, f_1 gives m_1, m_2 a different parity as always, and f_2 gives them the same value, since r, m_1, m_2 are all 1 in the index i (similar argument as before), so the sum of coefficients is 0 for each pair, and so for all coefficients corresponding to vectors in W.

This finishes the proof of (7), and so v_P is a linear combination of the vectors u_T. From the result above and Lemma 13 we deduce that G admits P-sparsification with error ε using $O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right)$ hyperedges, as required. □

To complete the picture for even k, we reduce to Proposition 14 by a simple "complementarity trick"

Proposition 15. *Let k be an even number, and G a k-uniform hypergraph. Let P be a k-ary Boolean predicate. Then for every $\varepsilon > 0$, G admits P-sparsification with error ε using $O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right)$ hyperedges.*

Proof. If $P(1, 1, \dots, 1) = 0$ then we are done by Proposition 14. Otherwise, we have that $P(1, 1, \dots, 1) = 1$, and consider $\overline{P} : \{0, 1\}^k \to \{0, 1\}$ defined by $\overline{P}(b_1, \dots, b_k) = 1 - P(b_1, \dots, b_k)$, so $v_{\overline{P}}$ is the negation of v_P. Since $\overline{P}$ has $\overline{P}(1, 1, \dots, 1) = 0$, Proposition 14 applies, and gives us a subhypergraph G_ε for each $\varepsilon > 0$, such that for every assignment $a : V \longrightarrow \{0,1\}$ we have

$$\left| \frac{|E|}{|E_\varepsilon|} \mathrm{Val}_{G_\varepsilon,\overline{P}}(a) - \mathrm{Val}_{G,\overline{P}}(a) \right| \leq \varepsilon(d_G|\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a)). \tag{8}$$

Now, since $v_P + v_{\overline{P}} = \mathbf{1}$ we have

$$\mathrm{Val}_{G,P}(a) + \mathrm{Val}_{G,\overline{P}}(a) = \langle v_P, v_{G,a} \rangle + \langle v_{\overline{P}}, v_{G,a} \rangle = \langle \mathbf{1}, v_{G,a} \rangle = |E|,$$

and the same is true for G_ε, and so we get

$$\mathrm{Val}_{G,\overline{P}}(a) = |E| - \mathrm{Val}_{G,P}(a) \tag{9}$$

and

$$\mathrm{Val}_{G_\varepsilon,\overline{P}}(a) = |E_\varepsilon| - \mathrm{Val}_{G_\varepsilon,P}(a). \tag{10}$$

Using (9) and (10) in (8), we get

$$\left| \frac{|E|}{|E_\varepsilon|} (|E_\varepsilon| - \mathrm{Val}_{G_\varepsilon,P}(a)) - (|E| - \mathrm{Val}_{G,P}(a)) \right| \leq \varepsilon(d_G|\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a)),$$

and after rearranging,

$$\left| \frac{|E|}{|E_\varepsilon|} \mathrm{Val}_{G_\varepsilon,P}(a) - \mathrm{Val}_{G,P}(a) \right| \leq \varepsilon(d_G|\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a)),$$

as required. □

The final piece in the jigsaw shows how to reduce sparsification of k-uniform hypergraphs with k odd to the case of $(k+1)$-uniform hypergraphs by adding a universal vertex and extending the original predicate by one dimension.

Proposition 16. *Let k be an odd number, and $G = (V, E)$ a k-uniform hypergraph. Let P be a k-ary Boolean predicate. Then for every $\varepsilon > 0$, G admits P-sparsification with error ε using $O\left(n\varepsilon^{-2} \log \frac{1}{\varepsilon}\right)$ hyperedges.*

Proof. Let $\varepsilon > 0$ and denote $\varepsilon_0 = \varepsilon \frac{k}{k+1}$. Consider the hypergraph $G' = (V', E')$ defined by

$$V' = V \cup \{v_0\} \quad , \quad E' = \{(v_1, \ldots, v_k, v_0) : (v_1, \ldots, v_k) \in E\},$$

where $v_0 \notin V$ is a new vertex. Clearly G' is a $(k+1)$-uniform hypergraph, and $k+1$ is even. We define a new predicate $P' : \{0,1\}^{k+1} \longrightarrow \{0,1\}$ by

$$P'(b_1, \ldots, b_{k+1}) = \begin{cases} 1 & P(b_1, \ldots, b_k) = 1, b_{k+1} = 1 \\ 0 & \text{otherwise} \end{cases}.$$

We may therefore apply Proposition 15 for G', P', ε_0 and deduce that for every assignment $a' : V' \longrightarrow \{0,1\}$ we have

$$\left| \mathrm{Val}_{G'_{\varepsilon_0},P'}(a') - \mathrm{Val}_{G',P'}(a') \right| \leq \varepsilon_0(d_{G'}|\mathrm{Z}_{a'}| + \mathrm{vol}_{G'}(\mathrm{Z}_{a'})) \tag{11}$$

for some subhypergraph $G'_{\varepsilon_0} = (V', E'_{\varepsilon_0} \subseteq E')$ which does not depend on a', and which satisfies $|E'_{\varepsilon_0}| = O\left(n\varepsilon_0^{-2} \log \frac{1}{\varepsilon_0}\right) = O\left(n\varepsilon^{-2} \log \frac{1}{\varepsilon}\right)$.

Let $G_{\varepsilon_0} = (V, E_{\varepsilon_0} \subseteq E)$ be the corresponding subhypergraph of G (so $|E_{\varepsilon_0}| = O\left(n\varepsilon^{-2} \log \frac{1}{\varepsilon}\right)$), and for every assignment $a : V \longrightarrow \{0,1\}$ define

$$a' : V' \longrightarrow \{0,1\} \quad , \quad a'(v) = \begin{cases} a(v) & v \in V \\ 1 & v = v_0 \end{cases}.$$

Since the hyperedge $e = (v_1, \ldots, v_k) \in E$ is counted in $\mathrm{Val}_{G,P}(a)$ if and only if the hyperedge $e' = (v_1, \ldots, v_r, v_0) \in E'$ is counted in $\mathrm{Val}_{G',P'}(a')$, we have that

$$\mathrm{Val}_{G,P}(a) = \mathrm{Val}_{G',P'}(a'),$$

and the same is true for G_{ε_0} and G'_{ε_0}. We get

$$\begin{aligned}
\left|\mathrm{Val}_{G_{\varepsilon_0},P}(a) - \mathrm{Val}_{G,P}(a)\right| &= \left|\mathrm{Val}_{G'_{\varepsilon_0},P'}(a') - \mathrm{Val}_{G',P'}(a')\right| \\
&\le \varepsilon_0 (d_{G'} |\mathrm{Z}_{a'}| + \mathrm{vol}_{G'}(\mathrm{Z}_{a'})) \\
&= \varepsilon_0 \left(\frac{(k+1)|E'|}{|V'|} |\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a) \right) \\
&= \varepsilon_0 \left(\frac{(k+1)|E|}{|V|+1} |\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a) \right) \\
&\le \varepsilon_0 \frac{k+1}{k} \left(\frac{k|E|}{|V|} |\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a) \right) \\
&= \varepsilon \left(d_G |\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a) \right),
\end{aligned}$$

where the second line is due to (11), the next two lines are by the definition of G' and a', the fifth line is by rearranging, and the last line is by the definition of the average degree of G. We get the required result. □

Propositions 15 and 16 complete the proof of Theorem 10.

Remark 17. In the proof of Proposition 14 the hypergraph G_ε was chosen independently of the predicate P. Since Propositions 15 and 16 reduce to that case, we have in fact shown that for every $\varepsilon > 0$, Theorem 10 is witnessed by the same subhypergraph G_ε for all different predicates P. This will be important in the proof of Theorem 21.

Remark 18. We note that our main result, Theorem 10, extends Theorem 3 in the regime where k is a constant, which is the main focus of this paper. However, Theorem 3 also works for non-constant k [7]. If k is not a constant, it can be seen from the proof of Lemma 13 that the number of hyperedges of the sparse subhypergraph is multiplied by a factor of $O(m^2)$ (since $O(m)$ is the proportion between ε and ε_0 given that the coefficients λ_i are constant). In Proposition 14 we have $m = 2^k$, and so for k not constant we get an additional factor of 4^k. Furthermore, in Propositions 8 and 14 we obtain extra factors of k, by considering the k-partite k-fold cover. While the regime with non-constant k is interesting for cuts, for arbitrary predicates one needs to be careful about representation as the natural (explicit) representation of (non-symmetric) predicates requires exponential space in the arity k.

Remark 19. As observed by one of the reviewers, our sparsification result (Theorem 10) actually shows sparsification under a stronger notion of sparsification, in which the right-hand side in (4) in Definition 9 is tighter. Namely, in the notation of Definition 9, we can require that

$$\left| \frac{|E|}{|E_\varepsilon|} \mathrm{Val}_{G_\varepsilon,P}(a) - \mathrm{Val}_{G,P}(a) \right| \le \varepsilon \min(d_G |\mathrm{Z}_a| + \mathrm{vol}_G(\mathrm{Z}_a), d_G |\mathrm{Z}_{a'}| + \mathrm{vol}_G(\mathrm{Z}_{a'})), \tag{12}$$

where $a'(v) = 1 - a(v)$ for every $v \in V$. In detail, Theorem 3 works for any assignment and thus in particular for a', the value of the Cut predicate is the same on a and a' (and thus also the left-hand side of (12) is the same for a and a'), and the rest is reductions that preserve (12).

4 SPARSIFICATION OF NON-BOOLEAN PREDICATES

We now focus on non-Boolean predicates; i.e., predicates of the form $P : D^k \to \{0,1\}$ with $|D| > 2$. Without loss of generality, we assume $D = [q]$ for some $q \geq 2$. The most natural way of generalising Theorem 10 to larger domains appears to be to use the same bound with $\mathrm{Z}_a = \{v \in V : a(v) = 0\}$. This, however, cannot give the desired sparsification result (cf. Section 5). Instead we use a different and somewhat weaker kind of generalisation of the Boolean case, and show that all hypergraphs are still sparsifiable with respect to all predicates using this definition.

Definition 20. Let $P : D^k \to \{0,1\}$ be a k-ary predicate where $D = [q]$. We say that a k-uniform hypergraph $G = (V, E)$ *admits all-but-one P-sparsification* with error ε using $O(f(n,\varepsilon))$ hyperedges if there exists a subhypergraph $G_\varepsilon = (V, E_\varepsilon \subseteq E)$ with $|E_\varepsilon| = O(f(n,\varepsilon))$ such that for every assignment $a : V \to D$ we have

$$\left| \frac{|E|}{|E_\varepsilon|} \mathrm{Val}_{G_\varepsilon,P}(a) - \mathrm{Val}_{G,P}(a) \right| \leq \varepsilon(d_G |M_a| + \mathrm{vol}_G(N_a)), \tag{13}$$

where M_a is the largest set among the sets $\{v \in V : a(v) = i\}$, N_a is the set with the largest volume among the sets $\{v \in V : a(v) = i\}$ for $0 \leq i \leq q-2$, and d_G is the average degree in G.

Observe that the maximum in Definition 20 is over $i = 0, \ldots, q-2$ without $i = q-1$, hence the name "all-but-one". We note that there is nothing special about $q-1$ and any value from $[q]$ could be chosen in Definition 20.

Under Definition 20, Theorem 10 generalises.

Theorem 21. *For every k-uniform hypergraph $G = (V, E)$, every k-ary predicate $P : D^k \to \{0,1\}$ with $D = [q]$ (k, q are constants), and every $\varepsilon > 0$, G admits P all-but-one sparsification with error ε using $O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right)$ hyperedges.*

Note that in the case of $q = 2$ we have $P : \{0,1\}^k \to \{0,1\}$, and Definition 20 and Theorem 21 coincide with Definition 9 and Theorem 10. This is because when $q = 2$ the definitions of M_a, N_a coincide with the definition of Z_a in the Boolean case.

In order to prove Theorem 21, we will generalise our notations from Section 3. For a k-ary predicate $P : D^k \to \{0,1\}$ we consider the vector $v_P \in \mathbb{R}^{q^k}$, defined by $v_P[i] = P(\mathrm{rep}_{q,k}(i))$. For a given hypergraph G and an assignment a, we consider the vector $v_{G,a} \in \mathbb{R}^{q^k}$ defined by $v_{G,a}[i] = \left|\{(v_1, \ldots, v_k) \in E : (a(v_1), \ldots, a(v_k)) = \mathrm{rep}_{q,k}(i)\}\right|$. In other words, each coordinate of $v_{G,a}$ counts the hyperedges in G whose vertices are assigned some specific set of values by a. Under these notations, just as before, we get $\mathrm{Val}_{G,P}(a) = \langle v_P, v_{G,a}\rangle$, where $\langle\cdot,\cdot\rangle$ is the standard inner product in $\mathbb{R}^{q^k}$.

We start by proving the result for *singleton predicates*, i.e. for predicates P such that $v_P = e_r$ for some $0 \leq r \leq q^k - 1$.

Lemma 22. *Let $G = (V, E)$ be a k-uniform hypergraph, and $P : D^k \to \{0,1\}$ a k-ary singleton predicate with $D = [q]$ (k, q are constants). For every $\varepsilon > 0$, G admits P all-but-one sparsification with error ε using $O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right)$ hyperedges.*

Proof. Denote $\varepsilon_0 = \frac{\varepsilon}{q}$. Let $\gamma(G) = (V^\gamma, E^\gamma)$ be the k-partite k-fold cover of G, and let $\gamma(G)_{\varepsilon_0} = (V^\gamma, E^\gamma_{\varepsilon_0} \subseteq E^\gamma)$ be the subhypergraph promised by Theorem 10. From Remark 17 we know that this is the same subhypergraph for all predicates, and it does not depend on the choice of P. As before, let $G_{\varepsilon_0} = (V, E_{\varepsilon_0} \subseteq E)$ be the subhypergraph of G which satisfies $\gamma(G_{\varepsilon_0}) = \gamma(G)_{\varepsilon_0}$.

Let r be the integer for which $v_P = e_r$, and denote $u_r = \mathrm{rep}_{q,k}(r)$. Consider the set $T = \{i \in [k] : u_r[i] = q-1\}$. For each assignment $a : V \to [q]$, we want to find a Boolean assignment

$a_r : V^\gamma \longrightarrow \{0,1\}$ which does not assign 0 to any vertices $v^{(i)}$ which have $a(v) = q-1$, but which can also be used to show P-sparsification. We define

$$a_r(v^{(i)}) = \begin{cases} 0 & i \notin T \text{ and } a(v) = u_r[i] \\ 0 & i \in T \text{ and } a(v) \neq q-1 \\ 1 & \text{otherwise} \end{cases}.$$

We also define a k-ary Boolean predicate $P_r : \{0,1\}^k \longrightarrow \{0,1\}$ to only have a single truth value (a singleton) which is $(b_1, \ldots, b_k)$ where $b_i = 0$ if $i \notin T$ and $b_i = 1$ otherwise.

Observe that

$$\mathrm{Val}_{G,P}(a) = \mathrm{Val}_{\gamma(G),P_r}(a_r), \tag{14}$$

since both count the same hyperedges. The same is true for any hypergraph, and specifically for G_{ε_0}, that is

$$\mathrm{Val}_{G_{\varepsilon_0},P}(a) = \mathrm{Val}_{\gamma(G_{\varepsilon_0}),P_r}(a_r). \tag{15}$$

Using Theorem 10 for $\gamma(G), P_r, \varepsilon_0$ and a_r we get

$$\begin{aligned}\left|\frac{|E|}{|E_{\varepsilon_0}|}\mathrm{Val}_{G_{\varepsilon_0},P}(a) - \mathrm{Val}_{G,P}(a)\right| &= \left|\frac{|E^\gamma|}{|E^\gamma_{\varepsilon_0}|}\mathrm{Val}_{\gamma(G_{\varepsilon_0}),P_r}(a_r) - \mathrm{Val}_{\gamma(G),P_r}(a_r)\right| \\ &\leq \varepsilon_0(d_{\gamma(G)}|\mathrm{Z}_{a_r}| + \mathrm{vol}_{\gamma(G)}(\mathrm{Z}_{a_r})) \\ &\leq \varepsilon_0\left(\frac{d_G}{k}\cdot kq|M_a| + q\cdot \mathrm{vol}_G(N_a)\right) \\ &= \varepsilon\left(d_G|M_a| + \mathrm{vol}_G(N_a)\right),\end{aligned}$$

where the first line follows from (14), (15) and Definition 6, the second line is the application of Theorem 10, the third is again Definition 6 and the definitions of a_r, M_a, N_a, and the last is the definition of ε_0. This is true for every assignment a. In addition we have

$$|E_{\varepsilon_0}| = O\left(n\varepsilon_0^{-2}\log\frac{1}{\varepsilon_0}\right) = O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right),$$

so G_{ε_0} is the required witness. □

The proof of Theorem 21 is now an application of Lemma 22 similar to the proof of Lemma 13.

Proof of Theorem 21. The vector v_P satisfies

$$v_P = \sum_{r=0}^{q^k-1}\lambda_r e_r,$$

for some $\lambda_r \in \{0,1\}$ and e_r vectors of the standard basis of $\mathbb{R}^{q^k}$. Therefore, for every assignment $a : V \longrightarrow D$, we have

$$\mathrm{Val}_{G,P}(a) = \langle v_P, v_{G,a}\rangle = \sum_{r=0}^{q^k-1}\lambda_r\langle e_r, v_{G,a}\rangle = \sum_{r=0}^{q^k-1}\lambda_r \mathrm{Val}_{G,P_r}(a) \tag{16}$$

where P_r is the predicate corresponding to the vector e_r. Let $G_{\varepsilon_0} = (V, E_{\varepsilon_0} \subseteq E)$ be the subhypergraph of G promised by Lemma 22 for $\varepsilon_0 = \frac{\varepsilon}{q^k}$. Note that this is the same subhypergraph for all predicates

P_r (see proof of the lemma). Equation (16) is true for any other hypergraph as well, and in particular G_{ε_0}. Using Lemma 22 for each P_r, we get

$$\begin{aligned}
\left|\frac{|E|}{|E_{\varepsilon_0}|}\mathrm{Val}_{G_{\varepsilon_0},P}(a) - \mathrm{Val}_{G,P}(a)\right| &= \left|\frac{|E|}{|E_{\varepsilon_0}|}\sum_{r=0}^{q^k-1}\lambda_r \mathrm{Val}_{G_{\varepsilon_0},P_r}(a) - \sum_{r=0}^{q^k-1}\lambda_r \mathrm{Val}_{G,P_r}(a)\right| \\
&\le \sum_{r=0}^{q^k-1}\lambda_r \left|\frac{|E|}{|E_{\varepsilon_0}|}\mathrm{Val}_{G_{\varepsilon_0},P_r}(a) - \mathrm{Val}_{G,P_r}(a)\right| \\
&\le \sum_{r=0}^{q^k-1}\lambda_r \varepsilon_0 (d_G|M_a| + \mathrm{vol}_G(N_a)) \\
&\le q^k \varepsilon_0 (d_G|M_a| + \mathrm{vol}_G(N_a)) \\
&= \varepsilon(d_G|M_a| + \mathrm{vol}_G(N_a)),
\end{aligned}$$

where the first line follows from (16) for the different hypergraphs, the second line from the triangle inequality, the third from Lemma 22, the fourth is due to $\lambda_r \in \{0,1\}$ for all r, and the last is the definition of ε_0. Again,

$$|E_{\varepsilon_0}| = O\left(n\varepsilon_0^{-2}\log\frac{1}{\varepsilon_0}\right) = O\left(n\varepsilon^{-2}\log\frac{1}{\varepsilon}\right),$$

so we have found an appropriate subhypergraph of G. □

Remark 23. Similarly to Remark 18, if k and q are not constant we get an additional factor of q^{2k}.

5 OPTIMALITY OF ALL-BUT-ONE SPARSIFICATION

One might wonder if there is a different, perhaps stronger way to define sparsification for predicates on non-Boolean domains. The following example shows that all-but-one sparsification is optimal.

For a hypergraph $G = (V, E)$ and a fixed assignment $a : V \longrightarrow [q]$ denote $S_i = \{v \in V : a(v) = i\}$ (so $S_0 = Z_a$). The definition of all-but-one sparsification lets us take a bound which depends on the sizes and volumes of all the sets S_i except for S_{q-1}. In fact, if we try to take a bound which depends on fewer of these sets, the definition fails to generalise even the most basic case of the Cut predicate. To see this, it is sufficient to consider the graph case, i.e. $k = 2$. Let us suppose, without loss of generality, that our bound does not depend on S_{q-2}, S_{q-1}. Consider the predicate $\mathrm{Cut} : [q]^2 \longrightarrow \{0,1\}$ defined by $\mathrm{Cut}(x,y) = 1 \iff x \neq y$. A simple (but lengthy) argument below shows that cliques do not have a Cut-sparsifier using such a definition. Therefore, no definition with a bound which depends on "less" is possible, under the current assumptions.

Let $G = K_n$ be the complete graph with n vertices $v_1, \ldots, v_n$. Moreover, let G_ε be a subgraph of G, and consider the predicate $\mathrm{Cut} : [q]^2 \longrightarrow \{0,1\}$ defined by $\mathrm{Cut}(x,y) = 1 \iff x \neq y$.

We consider two cases:

Case (1): All vertices in G_ε have the same degree $d > 0$.

We look at the assignment $a : V \longrightarrow \{0,1\}$ defined as $a(v_1) = a(v_2) = q-2$ and for all $i > 2$, $a(v_i) = q-1$. If G_ε is a Cut-sparsifier of G then

$$\left|\frac{|E|}{|E_\varepsilon|}\mathrm{Val}_{G_\varepsilon,\mathrm{Cut}}(a) - \mathrm{Val}_{G,\mathrm{Cut}}(a)\right| \le \varepsilon(d_G|S| + \mathrm{vol}_G(S)) = 0,$$

where S is some set which depends only on the empty sets $S_0, \ldots, S_{q-3}$. If v_1, v_2 are neighbours in G_ε this implies

$$\frac{n-1}{d} = \frac{\frac{n(n-1)}{2}}{\frac{nd}{2}} = \frac{|E|}{|E_\varepsilon|} = \frac{\mathrm{Val}_{G,\mathrm{Cut}}(a)}{\mathrm{Val}_{G_\varepsilon,\mathrm{Cut}}(a)} = \frac{2(n-2)}{2(d-1)} = \frac{n-2}{d-1},$$

and if they are not then

$$\frac{n-1}{d} = \frac{|E|}{|E_\varepsilon|} = \frac{\mathrm{Val}_{G,\mathrm{Cut}}(a)}{\mathrm{Val}_{G_\varepsilon,\mathrm{Cut}}(a)} = \frac{2(n-2)}{2d} = \frac{n-2}{d}.$$

The second option is a contradiction, and the first option implies $d = n-1$, which means $|E_\varepsilon| = |E|$, so G_ε is not a sparsifier.

Case (2): There exist two vertices v_i, v_j with degrees $d_i \neq d_j$ in G_ε.

We look at two assignments $a_1 : V \longrightarrow \{0,1\}$ defined by $a_1(v_i) = q-2$ and $a_1(v) = q-1$ for $v \neq v_i$, and $a_2 : V \longrightarrow \{0,1\}$ defined by $a_2(v_j) = q-2$ and $a_2(v) = q-1$ for $v \neq v_j$. Since

$$\frac{\mathrm{Val}_{G,\mathrm{Cut}}(a_1)}{\mathrm{Val}_{G_\varepsilon,\mathrm{Cut}}(a_1)} = \frac{n-1}{d_i} \neq \frac{n-1}{d_j} = \frac{\mathrm{Val}_{G,\mathrm{Cut}}(a_2)}{\mathrm{Val}_{G_\varepsilon,\mathrm{Cut}}(a_2)},$$

at least one side of the inequality is different from $\frac{|E|}{|E_\varepsilon|}$. Suppose without loss of generality this is the left hand side. Then

$$\left| \frac{|E|}{|E_\varepsilon|} Val_{G_\varepsilon,\mathrm{Cut}}(a_1) - Val_{G,\mathrm{Cut}}(a_1) \right| > 0 = \varepsilon(d_G|S| + \mathrm{vol}_G(S)),$$

where again S is some set not depending on S_{q-2}, S_{q-1}, and so G_ε is not a Cut-sparsifier of G.

Note that the same argument works for any predicate P with $P(q-2, q-1) = P(q-1, q-2) = 1$ and $P(q-2, q-2) = P(q-1, q-1) = 0$. Thus if a definition does not depend on more than just S_{q-2}, S_{q-1}, it specifically does not depend on these two, so the same argument still works.

ACKNOWLEDGMENTS

We would like to thank the anonymous referees of both the conference [29] and this full version of the paper. Stanislav Živný was supported by a Royal Society University Research Fellowship. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532). The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein. This work was also supported by UKRI EP/X024431/1.

REFERENCES

[1] Kook Jin Ahn and Sudipto Guha. 2009. Graph Sparsification in the Semi-streaming Model. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09), Part II (Lecture Notes in Computer Science, Vol. 5556)*. Springer, 328–338. https://doi.org/10.1007/978-3-642-02930-1_27

[2] Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. 1999. Fast Estimation of Diameter and Shortest Paths (Without Matrix Multiplication). *SIAM J. Comput.* 28, 4 (1999), 1167–1181. https://doi.org/10.1137/S0097539796303421

[3] Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. 1993. On sparse spanners of weighted graphs. *Discrete & Computational Geometry* 9, 1 (1993), 81–100. https://doi.org/10.1007/BF02189308

[4] Alexandr Andoni, Jiecao Chen, Robert Krauthgamer, Bo Qin, David P. Woodruff, and Qin Zhang. 2016. On Sketching Quadratic Forms. In *Proceedings of the 7th ACM Conference on Innovations in Theoretical Computer Science (ITCS'16)*. ACM, 311––319. https://doi.org/10.1145/2840728.2840753

[5] Baruch Awerbuch. 1985. Complexity of Network Synchronization. *J. ACM* 32, 4 (1985), 804–823. https://doi.org/10.1145/4221.4227

[6] Baruch Awerbuch, Yossi Azar, Avrim Blum, and Santosh S. Vempala. 1998. New Approximation Guarantees for Minimum-Weight k-Trees and Prize-Collecting Salesmen. *SIAM J. Comput.* 28, 1 (1998), 254–262. https://doi.org/10.1137/S009753979528826X

[7] Nikhil Bansal, Ola Svensson, and Luca Trevisan. 2019. New Notions and Constructions of Sparsification for Graphs and Hypergraphs. *Proceedings of the 60th IEEE Annual Symposium on Foundations of Computer Science (FOCS'19)* (2019), 910–928. https://doi.org/10.1109/FOCS.2019.00059

[8] Surender Baswana, Telikepalli Kavitha, Kurt Mehlhorn, and Seth Pettie. 2005. New constructions of (alpha, beta)-spanners and purely additive spanners. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'05)*. SIAM, 672–681. http://dl.acm.org/citation.cfm?id=1070432.1070526

[9] Surender Baswana and Sandeep Sen. 2007. A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Random Struct. Algorithms* 30, 4 (2007), 532–563. https://doi.org/10.1002/rsa.20130

[10] Joshua Batson, Daniel A. Spielman, and Nikhil Srivastava. 2014. Twice-Ramanujan Sparsifiers. *SIAM Rev.* 56, 2 (2014), 315–334. https://doi.org/10.1137/130949117

[11] András A. Benczúr and David R. Karger. 1996. Approximating s-t Minimum Cuts in $\tilde{O}(n^2)$ Time. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC'96)*. ACM, 47–55. https://doi.org/10.1145/237814.237827

[12] Béla Bollobás, Don Coppersmith, and Michael Elkin. 2005. Sparse Distance Preservers and Additive Spanners. *SIAM J. Discret. Math.* 19, 4 (2005), 1029–1055. https://doi.org/10.1137/S0895480103431046

[13] Richard A. Brualdi, Frank Harary, and Zevi Miller. 1980. Bigraphs versus digraphs via matrices. *Journal of Graph Theory* 4, 1 (1980), 51–73. https://doi.org/10.1002/jgt.3190040107

[14] Silvia Butti and Stanislav Živný. 2020. Sparsification of Binary CSPs. *SIAM J. Discret. Math.* 34, 1 (2020), 825–842. https://doi.org/10.1137/19M1242446

[15] Parinya Chalermsook, Syamantak Das, Bundit Laekhanukit, Yunbum Kook, Yang P Liu, Richard Peng, Mark Sellke, and Daniel Vaz. 2021. Vertex Sparsification for Edge Connectivity. In *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'21)*. https://arxiv.org/abs/2007.07862

[16] Shiri Chechik. 2013. New Additive Spanners. In *Proceedings of the 24th ACM-SIAM Symposium on Discrete Algorithms (SODA'13)*. SIAM, 498–512. https://doi.org/10.1137/1.9781611973105.36

[17] Edith Cohen. 1998. Fast Algorithms for Constructing t-Spanners and Paths with Stretch t. *SIAM J. Comput.* 28, 1 (1998), 210–236. https://doi.org/10.1137/S0097539794261295

[18] Dorit Dor, Shay Halperin, and Uri Zwick. 2000. All-Pairs Almost Shortest Paths. *SIAM J. Comput.* 29, 5 (2000), 1740–1759. https://doi.org/10.1137/S0097539797327908

[19] Michael Elkin and Ofer Neiman. 2020. Near-Additive Spanners and Near-Exact Hopsets, A Unified View. *Bull. EATCS* 130 (2020). http://bulletin.eatcs.org/index.php/beatcs/article/view/608/624

[20] Arnold Filtser and Robert Krauthgamer. 2017. Sparsification of Two-Variable Valued Constraint Satisfaction Problems. *SIAM J. Discret. Math.* 31, 2 (2017), 1263–1276. https://doi.org/10.1137/15M1046186

[21] Wai Shing Fung, Ramesh Hariharan, Nicholas J. A. Harvey, and Debmalya Panigrahi. 2019. A General Framework for Graph Sparsification. *SIAM J. Comput.* 48, 4 (2019), 1196–1223. https://doi.org/10.1137/16M1091666

[22] Torben Hagerup, Jyrki Katajainen, Naomi Nishimura, and Prabhakar Ragde. 1998. Characterizing Multiterminal Flow Networks and Computing Flows in Networks of Small Treewidth. *J. Comput. Syst. Sci.* 57, 3 (1998), 366–375. https://doi.org/10.1006/jcss.1998.1592

[23] Dmitry Kogan and Robert Krauthgamer. 2015. Sketching Cuts in Graphs and Hypergraphs. In *Proceedings of the 6th ACM Conference on Innovations in Theoretical Computer Science (ITCS'15)*. ACM, 367–376. https://doi.org/10.1145/2688073.2688093

[24] Rasmus Kyng, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Daniel A. Spielman. 2016. Sparsified Cholesky and multigrid solvers for connection laplacians. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC'16)*. ACM, 842–850. https://doi.org/10.1145/2897518.2897640

[25] Frank Thomson Leighton and Ankur Moitra. 2010. Extensions and limits to vertex sparsification. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC'2010)*. ACM, 47–56. https://doi.org/10.1145/1806689.1806698

[26] Ankur Moitra. 2009. Approximation Algorithms for Multicommodity-Type Problems with Guarantees Independent of the Graph Size. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS'09)*. IEEE Computer Society, 3–12. https://doi.org/10.1109/FOCS.2009.28

[27] Ilan Newman and Yuri Rabinovich. 2013. On Multiplicative Lambda-Approximations and Some Geometric Applications. *SIAM J. Comput.* 42, 3 (2013), 855–883. https://doi.org/10.1137/100801809

[28] David Peleg and Alejandro A Schäffer. 1989. Graph spanners. *Journal of graph theory* 13, 1 (1989), 99–116. https://doi.org/10.1002/jgt.3190130114

[29] Eden Pelleg and Stanislav Živný. 2021. Additive Sparsification of CSPs. In *Proceedings of the 29th Annual European Symposium on Algorithms (ESA'21) (LIPIcs, Vol. 204)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 75:1–75:15. https://doi.org/10.4230/LIPIcs.ESA.2021.75

[30] Liam Roditty, Mikkel Thorup, and Uri Zwick. 2005. Deterministic Constructions of Approximate Distance Oracles and Spanners. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05) (Lecture Notes in Computer Science, Vol. 3580)*. Springer, 261–272. https://doi.org/10.1007/11523468_22

[31] Tasuku Soma and Yuichi Yoshida. 2019. Spectral Sparsification of Hypergraphs. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*. 2570–2581. https://doi.org/10.1137/1.9781611975482.159

[32] Daniel A. Spielman and Nikhil Srivastava. 2011. Graph Sparsification by Effective Resistances. *SIAM J. Comput.* 40, 6 (2011), 1913–1926. https://doi.org/10.1137/080734029

[33] Daniel A. Spielman and Shang-Hua Teng. 2004. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC'04)*. ACM, 81–90. https://doi.org/10.1145/1007352.1007372

[34] Daniel A. Spielman and Shang-Hua Teng. 2011. Spectral Sparsification of Graphs. *SIAM J. Comput.* 40, 4 (2011), 981–1025. https://doi.org/10.1137/08074489X

[35] David P. Woodruff. 2010. Additive Spanners in Nearly Quadratic Time. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP'10) (Lecture Notes in Computer Science, Vol. 6198)*. Springer, 463–474. https://doi.org/10.1007/978-3-642-14165-2_40

Received 29 June 2021; revised 9 July 2023; accepted 19 September 2023