# Generation of Kochen-Specker contextual sets in higher dimensions by dimensional upscaling whose complexity does not scale with dimension and their applications

Mladen Pavičić [1,*] and Mordecai Waegell [2]

[1] *Center of Excellence for Advanced Materials and Sensing Devices (CEMS), Photonics and Quantum Optics Unit, Ruđer Bošković Institute and Institute of Physics, 10000 Zagreb, Croatia*

[2] *Institute for Quantum Studies, Chapman University, Orange, California 92866, USA*

(Received 16 July 2023; accepted 13 June 2024; published 3 July 2024)

Recently, handling of contextual sets, in particular Kochen-Specker (KS) sets, in higher dimensions has been given an increasing attention, both theoretically and experimentally. However, methods of their generation are diverse, not generally applicable in every dimension, and of exponential complexity. Therefore, we design a dimensional upscaling method, whose complexity does not scale with dimension. As a proof of principle we generate manageable-sized KS master sets in up to 27 dimensional spaces and show that well over 32 dimensions can be reached. From these master sets we obtain an ample number of smaller KS sets. We discuss three kinds of applications that work with KS sets in higher dimensions. We anticipate other applications of KS sets for quantum information processing that make use of large families of nonisomorphic KS sets.

DOI: 10.1103/PhysRevA.110.012205

## I. INTRODUCTION

It has been proven that applications in quantum computation [1,2], quantum steering [3], and quantum communication [4] rely on quantum contextuality and contextual sets. Small contextual sets, predominantly Kochen-Specker (KS) sets [5], in low dimensional spaces have been implemented in a series of experiments using photons [6–11], neutrons [12,13], trapped ions [14], and solid state molecular nuclear spins [15]. KS sets are sets that prove the KS theorem, i.e., sets of quantum measurements whose outcomes cannot be predicted by means of classical (noncontextual) value assignment [16].

Some of the applications that have been developed for KS sets, such as oblivious communication and quantum key distribution, can make use of any KS set, and having access to a large number of alternatives may actually increase the security of these protocols [4,17–22]. In this paper we give two applications that work with KS sets in any dimension, and another for a particular family of KS sets in even dimensions.

As for implementations of contextual sets, it is of interest to achieve them in any dimension. By employing photon orbital angular momentum technique one can implement "high-dimensional photon quantum gates" [23,24] in dimensions up to eight. Equally so for "multiphoton entanglement in high dimensions" [25] or for "time-energy entanglement with high-dimensional encoding" [26]. Apparently this is also a current upper limit for implementation of contextual sets [10,27,28]. However, the long-term goal of a quantum internet, quantum communication and quantum computation alike, requires implementations in much higher dimensions than current experiments can achieve; see [29] and [30], respectively.

For exploring the scope of future theoretical uses or experimental implementations of contextuality it would thus be valuable to develop a universal method of generating high dimensional contextual sets which may lead to new and different applications. Existing methods for finding contextual sets work in high dimensions are diverse and often limited to a particular family of sets, so the results are relatively few and are unevenly distributed. Let us first review some of these particular cases before moving on to our methods of generating families of contextual sets.

Ramanathan *et al.* [19] consider KS contextual sets in dimensions $\geqslant 2^{17}$ to obtain as large violations of noncontextuality inequalities as possible. Zhan and Hu [31] consider dimensions up to 20 to obtain "dimension-dependent noncontextuality inequalities." Frembs *et al.* [32] show that the contextuality of qudits of dimension $p^r$, $p$ prime, $r \in \mathbb{N}$, is a resource for a measurement-based quantum computation. Waegell and Aravind [33] show that codewords of the binary and ternary Golay codes can be converted into rays in $\mathbb{RP}^{23}$ and $\mathbb{RP}^{11}$ that provide proofs of the Kochen-Specker theorem in real state spaces of dimensions 24 and 12, respectively. Wang *et al.* [34] make use of $d^n$ dimensional unitary gate $U \in \mathrm{SU}(d^n)$ operating on the $n$-qudit state (where qudit is of dimension $d$) to provide a "high-dimensional quantum computing" resource and show that a contextual system solves a problem faster than the classical methods.

Many authors have therefore undertaken an effort to computationally generate contextual sets for possible subsequent usage and application; see [35–38] (Supplemental Material), and [39–43]. They generate KS sets from vectors, projectors, operators, graphs, stabilizer operations, polytopes, Lie groups, etc. Fortunately, all these entities can be reduced to hypergraphs.

So, in order to unify the generation methods and obtain more general results, exhaustive algorithms for obtaining

*Contact author: mpavicic@irb.hr

 

contextual hypergraphs from simple vector components, say, $\{0, \pm 1\}$, in odd and even dimensional spaces have recently been proposed [39,42]. But, although the algorithm of this method is valid for any dimension, in practice, it is limited by its exponential computational complexity. It faces a computational barrier in dimensions greater than eight.

Therefore, in this paper, we offer a method of generating manageable-sized master KS sets whose complexity does not grow with dimension and which can straightforwardly yield smaller sets, with implementation-optimal vector components $\{0, \pm 1\}$, in higher dimensional spaces. It can provide us with contextual sets of moderate size in a chosen dimension, each with distinct structural features that may be relevant for some particular application. We can generate new sets on demand limited only by computational resources, and we can also catalog and store many examples in a database for possible future usage. We stress here that some contextual sets in higher dimensions are so huge that most probably the need for very big ones will never materialize, but smaller sets, whose structural properties can be more easily understood, may be useful for many applications. Also, ample contextual sets in a chosen dimension might prove themselves indispensable for testing possible new hypotheses made of a set constructed in the dimension for the purpose.

The method stems from previous dimensional upscaling approaches [38,44–46] which offered several examples as proofs of principle for the methods, and fleshed out the simplest KS sets in lower dimensions. We extend and unify these approaches into a method for constructing KS sets with desired properties in any dimension, which enables us to find manageable-sized sets in higher dimensions using presently available computational resources. We give examples in all dimensions up to 16, then in 27D (27-dim), and finally we give a blueprint on how one can generate 32D examples. A generation of nonisomorphic sets in much higher dimensions is just a question of how much CPU time one is willing to dedicate to these tasks. The method relies on a remarkable feature of contextual sets that their "minimal complexity does not scale with dimension" as proved in [38]. Our results are discussed throughout the remainder of this paper.

To describe and handle KS sets we make use of McKay-Megill-Pavičić hypergraphs (MMPH) [43], which we will alternatively simply call *hypergraphs*. In Sec. II we elaborate on MMPHs in some detail. A hypergraph represents each state vector by a vertex, and states which are mutually orthogonal belong to *hyperedges*, which we will alternatively simply call *edges*. Hypergraphs are the most compact way to represent a KS set without omitting any structural details, and without assigning particular state vectors to the vertices (although some applications may depend in other ways on the particular assignment of state vectors).

The paper is organized as follows:

In Sec. II we present the hypergraph formalism we make use of, i.e., the MMPH formalism.

In Sec. III we introduce our dimensional upscaling method.

In Sec. IV we present master KS MMPHs we obtained by means of our dimensional upscaling method as well as the smaller KS MMPHs we obtained from the former MMPHs. Sections IV B, IV C, and IV F provide us with seeds for obtaining higher dimensional MMPHs in remaining subsections.

In Sec. V we offer three applications of higher dimensional MMPHs: Larger alphabet (Sec. V A), Oblivious communication protocol and communication of bounded-dimensional systems protocols (Sec. V B), and generalized Hadamard matrices (Sec. V B).

A discussion is given in Sec. VI.

In the Appendix we give KS MMPH′s strings and coordinatizations of masters and the smallest KS MMPHs for each dimension obtained in Sec. IV.

## II. HYPERGRAPH FORMALISM

An MMPH is a special case of a hypergraph. An $n$D ($n$-dim) MMPH is a connected *hypergraph $k$-$l$* with $k$ vertices and $l$ hyperedges (often simply called edges) in which (i) every vertex belongs to at least one hyperedge; (ii) every hyperedge contains $n$ vertices; (iii) no hyperedge shares only one vertex with another hyperedge; (iv) hyperedges may intersect each other in at most $n-2$ vertices; and (v) graphically, vertices are represented as dots and hyperedges as (curved) lines passing through them.

We encode MMPHs by means of the printable ASCII characters for each vertex, with the exception of "space," "0," "+," "," and ".". When all 90 characters are exhausted, we reuse them prefixed by "+" (again for each vertex), when those are exhausted by "++," and so on. Hyperedges are separated by "," and each MMPH is terminated by ".". There is no limit on their size.

A KS MMPH is an $n$D ($n \geqslant 3$) MMPH to whose vertices it is impossible to assign 1s and 0s in such a way that the following rules hold: (i) no two vertices in any edge are both assigned the value 1 and (ii) in no edge all of the vertices are assigned the value 0 [43], Theorem 3.2.

A given MMPH may or may not have a coordinatization, i.e., a representation (of vertices) by means of vectors in a Hilbert space.

Our notion of coordinatization differs the notion *orthonormal representation* used by Lovász [47]. First, Lovász considers an orthonormal representation of unit vectors in a Euclidean space such that if $i$ and $j$ are nonadjacent [not connected by an edge] vertices, then $\mathbf{v}_i$ and $\mathbf{v}_j$ are orthogonal, while in our notation vectors assigned to adjacent [connected by an (hyper)edge] vertices are orthogonal. Second, to Lovász every graph has an orthonormal representation, while not every MMPH has a coordinatization, e.g., the 6-3 KS MMP (hyper)graph shown in [39], Fig. 1, does not have it.

When a coordinatization is attached to vertices of a KS MMPH, then the KS theorem [5,44] states that such a hypergraph exists. This is in contrast to classical systems (in, e.g., classical computation) which always allow the aforementioned assignments of 1′s and 0′s.

For a KS MMPH with a coordinatization, its $n$D space becomes an $n$D Hilbert space spanned by $n$-tuples of mutually orthogonal vectors, where the $n$-tuples correspond to hyperedges and vectors to vertices.

A $k$-$l$ MMPH *class* is a collection of all sub-MMPHs contained in the $k$-$l$ MMPH. (A class may contain both non-KS and KS MMPHs.)

A *critical* KS MMPH is a KS MMPH which is minimal in the sense that removing any of its hyperedges turns it into a non-KS noncontextual MMPH.

A KS MMPH *master* is a noncritical KS MMPH which contains smaller KS proper sub-MMPHs.

The smallest masters contain just one critical proper KS MMPH. A master may contain non-KS MMPH. A master must contain at least one proper KS sub-MMPH.

A classical vertex index, $HI_c$, is the number of 1s one can assign to vertices of an MMPH so as to satisfy conditions (i) and (ii) of the KS theorem. Maximal (minimal) $HI_c$ is denoted as $HI_{cM}$ ($HI_{cm}$). It can be proved that $HI_{cM} = \boldsymbol{\alpha}$ [43, Th. 3.2], where $\boldsymbol{\alpha}$ is Lovász′s independence number [48, p. 192]. $HI_{cm}$ enables us to visually and/or numerically straightforwardly prove any KS MMPH (see Sec. IV L).

### III. DIMENSIONAL UPSCALING METHOD

An essential feature of any set we consider is that it consists of mutually orthogonal elements organized in blocks that are themselves mutually linked so as to form the set. These elements might be operators, projectors, states, vectors, graph vertices, or hypergraph vertices; e.g., vector blocks in an $n$-dim space are $n$-tuples of mutually orthogonal vectors. For all these elements and blocks we adopt an MMPH representation which we shall often simply call a hypergraph representation. To arrive at our *master* sets below we make use of vector representation, i.e., of hypergraphs with coordinatization. To generate smaller sets from master sets and therefore to form their distributions we make use of hypergraphs without a coordinatization since then their handling and computation are much faster and data much more compact. They reacquire a coordinatizations in a separate step after generation.

In [39,43] we generated billions of KS hypergraphs in dimensions up to eight, directly from simple vector components. Such a generation in 9+ dimensional spaces takes too much CPU time even on supercomputers, though.

To generate comparatively small KS hypergraphs in dimensions higher than those obtained directly from vector components in [39,43], we make use of a generalization of methods developed in [37–39,42,43].

We generalize the Matsuno/Penrose-Zimba method, along with some other tools, to generate manageable-sized KS hypergraphs which could then be searched for smaller KS sub-hypergraphs they contain. Our generalization works by combining two KS sets from lower dimensions so as to allow the number of unique vertices in the new combined set to be minimized; it is not guaranteed that the new set gives a KS hypergraph, so this has to be checked as a separate step. Given a KS set of dimension $n_1$ with $k_1$ vectors and another of dimension $n_2 \leqslant n_1$ with $k_2$ vectors, we can construct a new set in any dimension $n \leqslant n_1 + n_2$. The most interesting cases are with $n < n_1 + n_2$, since then the number of resulting unique vectors may be significantly less than $k_1 + k_2$.

The method works by extending the vectors of each parent set with enough vector components 0 to reach dimension $n$. For the first set, these are all appended at the end of the existing vectors. For the second set, the $n - n_2$ vector components 0 are distributed among the first $n_1$ dimensions (the same placements for every vector in the second set), so that the last $n - n_1$ dimensions are occupied by the nonzero vector components of the second set, which ensures that the new set has nonzero vector components in all $n$ cardinal directions in the space. At this point, some of the new vectors from the first set may be identical to some of the new vectors from the second set, which reduces the total number of unique vectors in the new set. Thus, to find the new set with the fewest unique vectors, we consider all permutations of the locations of the vector components 0 in the second set, and for each of these, all permutations of the order of the $n_2$ dimensions. Because we are using the vector components $\{0, \pm 1\}$, this method allows us to find new sets with $k$ significantly less than $k_1 + k_2$ for many choices of parent KS sets, and nearly all of these turn out to give KS hypergraphs—which might not work for more complicated sets of vector components.

Once we find a minimal set we can check whether it gives a KS hypergraph or not. If it does, we have a new master set, which likely contains many smaller KS hypergraphs that we can search for using other methods.

To use the new sets as parents in another round of upscaling to higher dimension, we want the new KS sets with smallest number of vectors, so we also check whether any particular vertices can be removed, along with all of their associated edges, to leave a smaller KS hypergraph behind, and in many cases this allows us to remove several extraneous vectors.

Finding new KS sets proceeds in a cyclic way, combining known KS set in small dimensions to find the new ones in slightly higher dimensions, keeping the smallest ones, and then repeating this process and moving to higher dimensions at each iteration. We probably have not found the smallest sets with vector components $\{0, \pm 1\}$ for dim $> 8$, but it seems likely that we do have them for $4 \leqslant \text{dim} \leqslant 8$.

Finding the KS sets with the fewest vertices is also important for computing their set of edges, which takes time that scales exponentially with both the dimension and the rough number of vertices in the set.

Our general goal was to find new master KS hypergraphs with a relatively small number of edges (ideally around 100, but not more than 1000), which could then be effectively searched for many smaller KS hypergraphs using our programs MMPSTRIP, MMPSHUFFLE, and STATES01 [43] with presently available computational resources. The results of these searches are presented and discussed in Secs. IV and VI.

In particular, this allows us to find the KS hypergraphs with fewest edges, and the general pattern that emerges is consistent with the general result that the minimum complexity of KS hypergraphs does not scale with dimension. As expected, the minimum number of vertices seems to grow roughly linearly with dimension, while the minimum number of edges in $\geqslant$4D fluctuates between 9 and 16. With different vector components, KS sets with fewer edges are known in certain dimensions (recall that among the 6D hypergraphs with complex vector components the smallest one has just seven edges [49]), so the general minimum is not known in all dimensions and for all vector components, but this gives an upper bound.

We present and discuss the outcomes of our method in all dimensions up to 16 and in 27D in Sec. IV.

## IV. HYPERGRAPHS GENERATED BY THE UPSCALING METHOD IN HIGHER DIMENSIONS

We generate MMPHs with coordinatizations with vector components from the set $\{0, \pm 1\}$. Their figures and distributions are shown in this section, and their strings and coordinatizations are given in the Appendix.

The first three subsections below we provide for the sake of completeness and because we use some of their MMPHs as seeds for generating MMPHs in higher dimensions via dimensional upscaling in subsequent subsections.

### A. 3D MMPHs—Vector component generation

3D MMPH generated from $\{0, \pm 1\}$ is not a KS set [42].

### B. 4D MMPHs—Vector component generation

We obtain the KS master MMPH 24-24 directly from the vector component set $\{0, \pm 1\}$ via our programs at [50], VECFIND (which determines generates a master from vector components and/or whether a vector assignment to vertices in an MMPH is possible), MMPSTRIP (which outputs all subsets of the input MMPHs that have a specified number of hyperedges removed), STATES01 (which determines whether an MMP diagram admits a 0,1 (nondispersive) state), and SHORTD (which removes duplicates from input MMPHs): (`vecfind -4d -master -nommp -vgen=0,1,-1 | mmpstrip -U | grep v24e24| states01 -1 -r 1000 | shortd -G`). From it we obtain six critical KS MMPHs, shown in Fig. 1(a) (two 20-11 and two 22-13) within seconds on a single CPU. Also, via our program MMPSUBSET, we obtain $2^{24} - 1$ sub-MMPHs from which we filter out 1232 nonisomorphic KS MMPHs via STATES01 and SHORTD within minutes on a single CPU [51]. Previously, the 24-24 and 18-9 were obtained in [52] and [53], respectively, by other methods. See also a diagrammatic representation of 24-24 [54]. A graphical presentation of 18-9 was first given in [55], Fig. 3(a), with redundant cyclically closed hyperedges. In this paper, all MMPHs are presented by means of non-redundant graphical presentation of hyperedges. For

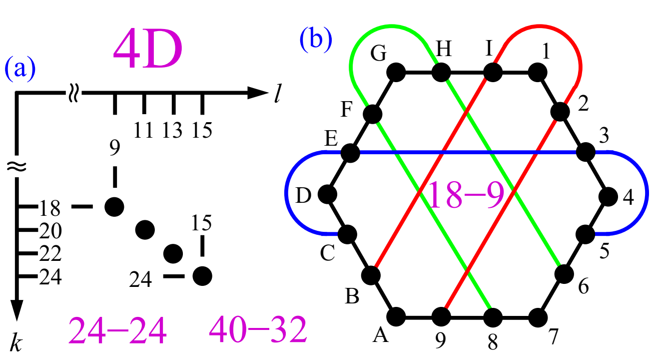


FIG. 1. (a) Distribution of the 4D critical KS MMPHs obtained from the master 40-32 itself generated by $\{0, \pm 1\}$ vector components [39]; 40-32 consists of two MMPHs: a KS 24-24 and a noncontextual 16-8; abscissa is $l$ (number of hyperedges); (negative) ordinate is $k$ (number of vertices). (b) The smallest 4D critical KS MMPH 18-9; $HI_{cM} = \boldsymbol{\alpha} = 4$, $HI_{cm} = 3$; strings and coordinatizations of 24-24, 18-9 are given in the Appendix 1.

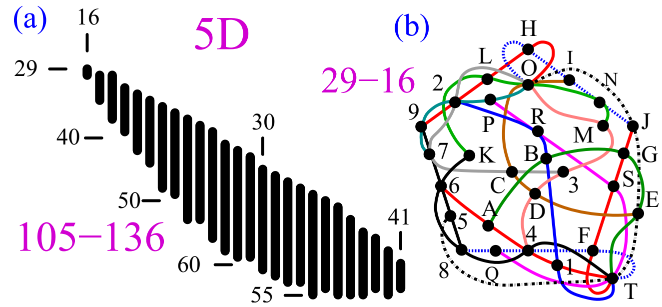


FIG. 2. (a) Distribution of the 5D critical KS MMPHs obtained from the master 105-136 itself generated by $\{0, \pm 1\}$ vector components [42]; abscissa is $l$ (number of hyperedges); (negative) ordinate is $k$ (number of vertices); cf. Fig. 1(a). (b) One of the two smallest five-dimensional 29-16 critical KS MMPH; $HI_{cM} = \boldsymbol{\alpha} = 7$, $HI_{cm} = 3$; strings and coordinatizations are given in the Appendix 2.

instance, in Fig. 1(b) vertices `C` and `5` are not connected directly by a straight line since they are already connected via the `CE34` line.

In the subsequent paragraphs we make use of these three KS MMPHs as well as of some higher dimensional MMPHs as seeds for generating MMPHs via dimensional upscaling in higher dimensions.

### C. 5D MMPHs—Vector component generation

Arbitrarily exhaustive number of critical 5D KS MMPHs shown in Fig. 2(a) were obtained in [41,42] directly from the set $\{0, \pm 1\}$ by vector component generation. Critical 29-16 [Fig. 2(b)] serves us as a seed for generating some master MMPHs in higher dimensions.

### D. 6D MMPHs—Dimensional upscaling

Six dimensional *Hilbert spaces* are inhabited by either spin-$\frac{5}{2}$ systems or by qubit-qutrit systems ($\mathcal{H}^6 = \mathcal{H}^2 \otimes \mathcal{H}^3$). The former representation in a complex space has been implemented in [11]. It turned out to have a fairly small master MMPH [39,41,56], and therefore it had been easy to find all its subgraphs. In the real space, the 332-1408 master MMPH generated by $\{0, \pm 1\}$ components is huge, and already at the time of its generation [37,39] we were well aware that the obtained 34-16 critical might not be the smallest. Our dimensional upscaling confirms that conjecture and provides us with a 31-**15** critical MMPH shown in Fig. 3(b). We obtain it from the master 31-16. From another 33-19 master we obtain three additional MMPHs, one of which (32-16) is also smaller then the 34-16. Their distributions are shown in Fig. 3(a).

### E. 7D MMPHs—Dimensional upscaling

The vector components $\{0, \pm 1\}$ in a 7D spin-3 space generate an 805-9936 master MMPH, which generates a class whose partial distribution is shown in [42]. Its vectors were automatically generated from the master MMPH by means of our programs MMPSTRIP, MMPSHUFFLE, and STATES01.

However, such a direct exhaustive generation takes too much CPU time on a supercomputer, and therefore we here generate a partial distribution of KS MMPHs with small num-

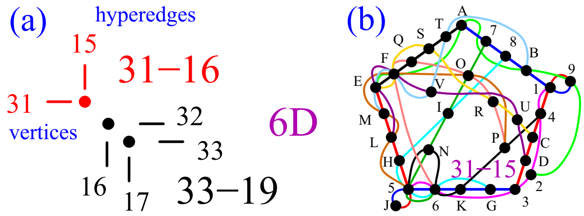


FIG. 3. (a) Distribution of the smallest 6-dim critical KS MMPHs obtained from the masters 31-16 and 33-19 by dimensional upscaling; abscissa is $l$ (number of hyperedges); (negative) ordinate is $k$ (number of vertices); cf. Fig. 1(a). (b) The smallest critical KS MMPHs 31-15 in the 31-16 class; $HI_{cM} = \boldsymbol{\alpha} = 7$, $HI_{cm} = 2$; strings and coordinatizations of 31-15, 32-16, and 33-17 are given in the Appendix 3.

ber of vertices and hyperedges (not obtained in [42]) from master 47-176 obtained here via dimensional upscaling as shown in Fig. 4(a).

The smallest critical 34-**14** MMPH from this class is shown in Fig. 4(b). Note the cutoff level at the 63 vertices characteristic of all masters that are not generated directly from the vector components. The ones that are we sometimes call *supermasters* and the former ones a *submasters*. Analogous cutoffs are evident for all higher dimensional submasters below.

#### F. 8D MMPHs—Vector component generation

Billions of 8D critical KS MMPHs obtainable from the 3280-1361376 $\{0, \pm\}$ master are given in [37]. So there is no need to employ the dimensional upscaling to replicate some of them here. However, we do make use of the smallest 34-**9** KS MMPH, shown in Fig. 5(a), as a seed for dimensional upscaling of MMPHs in higher dimensions.

#### G. 9D MMPHs—Dimensional upscaling

Two entangled qutrits live in a 9D space, and in [43] we generated their MMPH supermaster from $\{0, \pm 1\}$ components. It consists of 9586 vertices and 12 068 704 hyperedges, and that is too huge for a direct generation from the master MMPH. However, small critical KS MMPHs can be obtained by dimensional upscaling, which yields 47-144 and 63-1200 master MMPHs, whose distributions are shown in Fig. 6(a). Critical KS MMPHs are generated from the masters via STATES01.

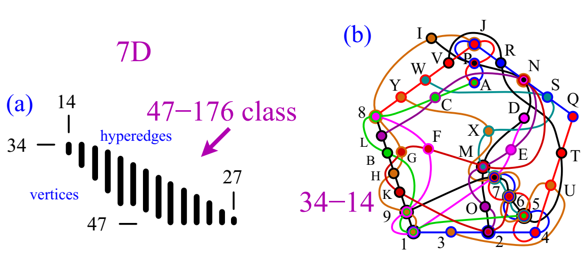


FIG. 4. (a) Distribution of a million of 7D critical KS MMPHs from the 47-176 master; cf. Fig. 1(a). (b) The smallest criticals 34-14 from the 47-176 master; $HI_{cM} = \boldsymbol{\alpha} = 7$, $HI_{cm} = 3$; strings and coordinatizations of 33-14 are given in the Appendix 4.

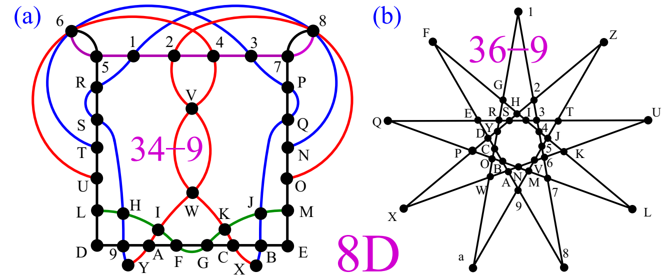


FIG. 5. (a) One of the two smallest nonisomorphic criticals 34-9 [37]; $HI_{cM} = \boldsymbol{\alpha} = 4$, $HI_{cm} = 3$; cf. Fig. 1(a). (b) Starlike 36-9 [37] proved by the $S$-$H$ theorem ([57]; see Sec. V C); strings and coordinatizations are given in the Appendix 5.

#### H. 10D MMPHs—Dimensional upscaling

From the spin-$\frac{9}{2}$ MMPH masters 52-141, 60-96, and 74-610, whose distributions are shown in Fig. 7(a), we obtain critical MMPHs, the smallest of which (50-**15**) is shown in Fig. 7(a). Their strings might have gaps in characters (e.g., 3 and K are missing in 50-15). The gaps can be closed by program MMPSHUFFLE if needed (e.g., for further processing).

The 10D KS master MMPHs are generated by means of particular combinations of hypergraphs from smaller dimensions (see Sec. IV J).

#### I. 11D MMPHs—Dimensional upscaling

Spin-5 11D master MMPHs with $\{0, \pm 1\}$ vector components generate smaller critical KS MMPHs by the dimensional upscaling. The master MMPHs are 50-38, 54-162, 65-198, and 71-4224. From them we generated the following critical KS MMPHs: 50-14, ..., 60-16, 54-18, ..., 54-23, and 61-21,..., 70-30 (each of which is coming in many nonisomorphic instances) whose distributions are shown in Fig. 8(a). We give the smallest one 50-**14** in Fig. 8(b).

#### J. 12D MMPHs—Dimensional upscaling

12D MMPHs can be represented either by spin-$\frac{11}{2}$ systems or by two qubits and a qutrit ($\mathcal{H}^{12} = \mathcal{H}^2 \otimes \mathcal{H}^2 \otimes \mathcal{H}^3$). The critical 52-**9** shown in Fig. 9(b) is one of 502

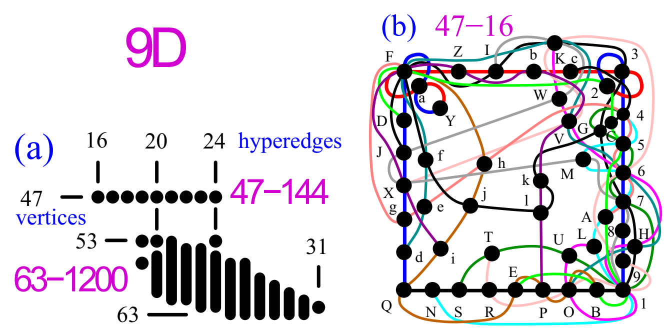


FIG. 6. (a) Distribution of 41 000 9D critical KS MMPHs obtained via dimensional upscaling from the 47-144 and 63-1200 master MMPHs; cf. Fig. 1(a). (b) The smallest critical 47-16 from the 47-144 class; $HI_{cM} = \boldsymbol{\alpha} = 7$, $HI_{cm} = 3$; strings and coordinatizations of 47-16 are given in the Appendix 6.

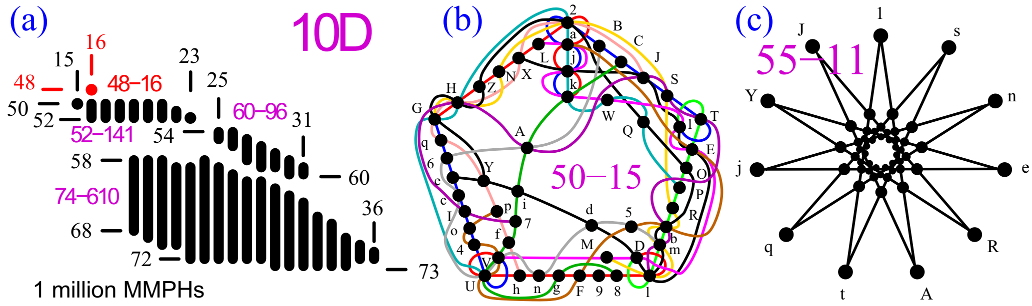


FIG. 7. (a) Distribution of 10D critical KS MMPHs obtained from the 52-141, 60-96, and 74-610 master MMPHs; cf. Fig. 1(a). (b) The smallest critical KS MMPH 50-15 from the 52-141 class; $HI_{cM} = \boldsymbol{\alpha} = 7$, $HI_{cm} = 2$. (c) Starlike KS MMPH 55-11 whose existence was proved by the *S*-*H* theorem ([57]. see Sec. V C. The string and coordinatization of the 50-15 and the string of the 55-11 are given in the Appendix 7; it is a critical KS MMPH with a parity proof.

nonisomorphic critical 52-9s from the 52-81 master. It represents a partial constructive proof (for MMPHs with $\{0, \pm 1\}$ component coordinatization) of the result that MMPHs in an even dimensional space $n \geqslant 10$ require at most nine hyperedges [38].

The 12D 52-81 master was obtained by combining the 34-9 set in 8D with the 18-9 in 4D. The 67-419 master is obtained as follows. First, we combine two 18-9 sets in 4D to get a 35-32 in 7D. Next, we combine the 35-32 in 7D with the 18-9 in 4D to get a 52-141 set in 10D [see Fig. 7(a)]. And finally, we combine this 52-141 set in 10D with the 18-9 in 4D to get a 67-419 in 12D. Similarly with the remaining four masters. We then apply MMPSTRIP and STATES01 programs to them to obtain distributions of all smaller KS hypergraphs contained in the masters. The five distributions are shown in the figure. The smallest KS hypergraph 52-9 [shown in Fig. 9(b)] is contained in just one of the masters (52-81).

### K. 13D MMPHs—Dimensional upscaling

Spin-6 13D criticals MMPHs shown in Fig. 10 are less abundant than the ones obtained in lower dimensions. This is due to the complexity of generation of MMPHs which forced us to generate as small masters as possible in order to shorten the runtimes of generation. However, although the number of vertices and hyperedges is limited we still get a high number of nonisomorphic criticals. Actually, among 100 000 of generated MMPHs there are no two isomorphic ones. Their distributions are shown in Fig. 10(a). The smallest critical is 63-**16** [Fig. 10(b)].

### L. 14D MMPHs—Dimensional upscaling

Spin-$\frac{13}{2}$ 14D criticals MMPHs shown in Fig. 11(a) are even fewer than the 13D ones, again due to the generation complexity. Longer CPU time is not viable at the present level of our research which is to show that one can generate thousands of comparatively small nonisomorphic criticals in high dimensional spaces. The smallest critical we obtained is 66-**15** in Fig. 11(b).

$HI_{cm} = 2$ for 66-15 means that it is, for instance, possible to assign 1 to just `b` and `B`. Then all other vertices must be assigned 0 and that violates the condition (ii) of the KS theorem for, e.g., the top red hyperedge. Alternatively, a numerical verification can be carried out on its MMPH string (Appendix 11) proving that the (top red) hyperedge `ekdfghijlm47E3` contains neither `b` nor `B`.

### M. 15D MMPHs—Dimensional upscaling

With spin-7 15D criticals MMPHs shown in Fig. 12 we succeeded in establishing an optimal dimensional upscaling. So the complexity of generation just allowed a more abundant distribution than in the 14D space, while the minimal MMPH

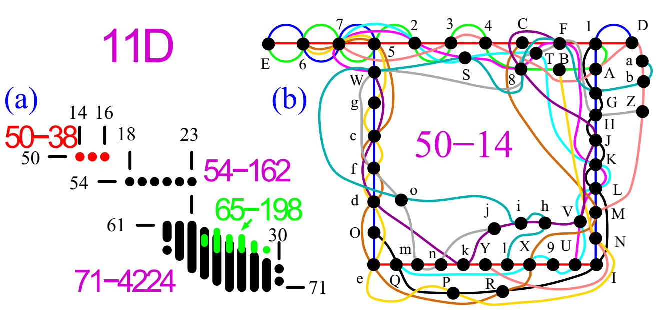


FIG. 8. (a) Distribution of 11D critical KS MMPHs obtained from the 50-38, 54-162, 65-198, and 71-4224 master MMPHs via dimensional upscaling; cf. Fig. 1(a). (b) The smallest critical 50-14 we obtained from the 50-38; $HI_{cM} = \boldsymbol{\alpha} = 6$, $HI_{cm} = 3$; strings and coordinatizations of 50-14 are given in the Appendix 8.

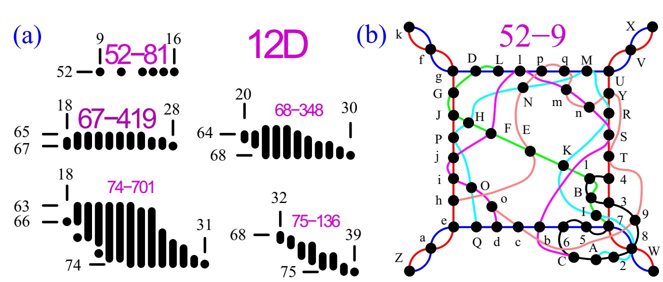


FIG. 9. (a) Distribution of 12D critical KS MMPHs obtained from the 52-81, 67-419, 68-348, 74-701, and 75-136 master MMPHs which are themselves obtained via dimensional upscaling; cf. Fig. 1(a). (b) The smallest critical KS MMPH 52-9 we obtained from the 52-81 master; $HI_{cM} = \boldsymbol{\alpha} = 4$, $HI_{cm} = 3$; strings and coordinatizations of 52-9 are given in the Appendix 9.

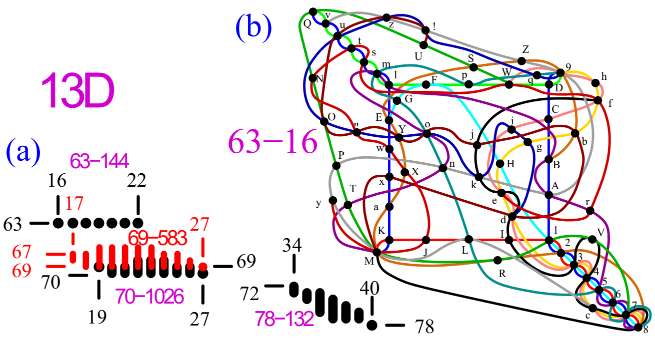


FIG. 10. (a) Distribution of 100 000 13D critical KS MMPHs obtained from the 63-144, 69-583, 70-1026, and 78-132 master MMPHs obtained via dimensional upscaling; cf. Fig. 1(a). (b) One of the smallest criticals, KS MMPH 63-16, we obtained from the 63-144 master; $HI_{cM} = \boldsymbol{\alpha} = 7$, $HI_{cm} = 3$; strings and coordinatizations of 63-16 are given in the Appendix 10.

has fewer hyperedges. The advantage of the low-level distribution is that we obtain fairly small criticals. Note that the smallest obtained MMPH in the 15D space—66-**14** shown in Fig. 12(b)—has fewer hyperedges than the ones in 13D and 14D obtained above; in the 16D they are even fewer.

### N. 16D MMPHs—Dimensional upscaling

16D space hosts four entangled qubits: $n = 2^4$. The smallest 16D critical MMPH with nine hyperedges 70-**9** we obtained via dimensional upscaling from the master 80-855, shown in Fig. 13(b), confirms our result from [38] according to which 4$n$D MMPHs (for positive integers $n$) require at most nine hyperedges, and at most 15 hyperedges in dimensions 4n+2.

### O. 27D MMPHs—Dimensional upscaling

The 27D is the space of three entangled qutrits. We apply a rank-scaling method to our 9D 47-16 to get a 27D 141-16 ($3 \times 47 = 141$) shown in Figs. 13(c) and 13(d) (its hyperedges are too interwoven to be discernible in (d); therefore, only its biggest loop is given). According to the hyperedge pattern that other MMPHs follow we should be able to obtain

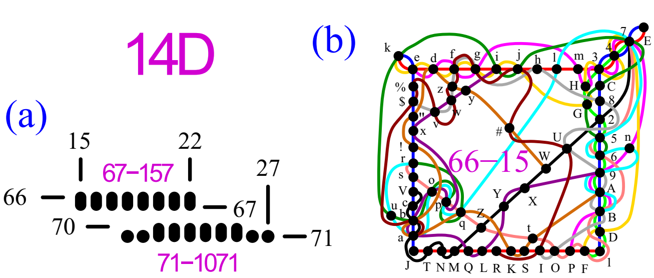


FIG. 11. (a) Distribution of 20 000 14D critical KS MMPHs obtained from the 67-157 and 71-1071 master MMPHs obtained via dimensional upscaling; cf. Fig. 1(a). (b) The smallest critical KS MMPH 66-15 we obtained from the 67-157 master; $HI_{cM} = \boldsymbol{\alpha} = 6$, $HI_{cm} = 2$; strings and coordinatizations of 66-15 are given in the Appendix 11.

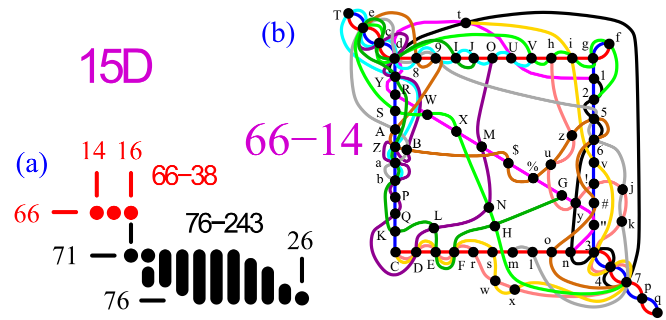


FIG. 12. (a) Distribution of 80 000 15D critical KS MMPHs obtained from the 66-38 and 76-243 master MMPHs; cf. Fig. 1(a). (b) The smallest critical KS MMPH 66-14 we obtained from the 66-38 master; $HI_{cM} = \boldsymbol{\alpha} = 6$, $HI_{cm} = 2$; strings and coordinatizations of 66-14 are given in the Appendix 12.

a 27D MMPH with 14 hyperedges if we only kept our computer search running long enough.

### P. 32D MMPHs—Dimensional upscaling

We are also able to straightforwardly generate 32D (five entangled qubits) by combining two of the 16D 68-9s to get 32D 136-9 but did not do it here because the hypergraph is already listed in [38], Table V, under $n = 8N$, $34N$-9 and because quite a number of 32D critical KS hypergraphs, obtained by another method, is already given in [37], Sec. XI.

## V. APPLICATIONS

Exhaustive development of the applications presented here is outside of the scope of this paper. However, we demonstrate that any and all KS sets in the higher dimensions presented below do have potential practical usage.

### A. Larger alphabet

We extend the lager alphabet procedure from [58]. A 4D KS "protection" of quantum key distribution (QKD) protocols, has been put forward in [59] based on a modification of the BB84 protocol in [60]. A KS hypergraph with nine edges has been used. (i) Alice randomly picks one of nine edges

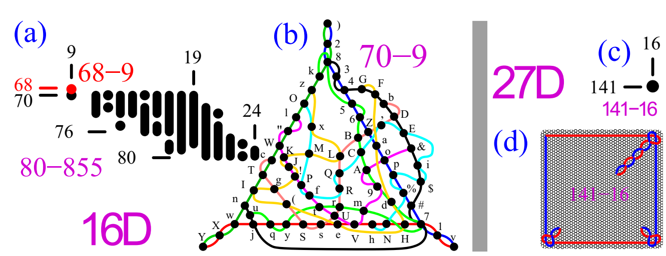


FIG. 13. (a) Distribution of 10 000 16D critical KS MMPHs obtained from the 80-855 master MMPH; cf. Fig. 1(a). (b) The smallest critical KS MMPH 70-9 we obtained from the master 80-855; $HI_{cM} = \boldsymbol{\alpha} = 4$, $HI_{cm} = 3$. (c) Critical 141-16 is the only MMPH of the 141-16 master. (d) Its biggest loop. $HI_{cM} = \boldsymbol{\alpha} = 7$, $HI_{cm} = 3$ strings and coordinatizations of 70-9 and 141-16 are given in the Appendix 13.

(bases) and sends Bob a randomly chosen state (vertex) of that edge; (ii) Bob picks one of nine edges at random and measures the system received from Alice. So, instead of qubits, we deal with ququarts and the information transferred via a system is not one but two bits [58]. We can modify and generalize this QKD protocol so as to apply it to any $k$-$l$ hypergraph ($k$ vertices, $l$ edges). The probability of Bob's picking a correct edge is $\frac{1}{l}$ and the number of vertices rises linearly with dimension (see Table I). The higher the dimension, the more errors Eve introduces. However, the main advantage of the KS QKD is that the number of 1's one can assign to vertices is maximized [43], Def. 4.7, Lemma 4.8 (Lovász's independence number $\boldsymbol{\alpha}$ [48], p. 192). $\boldsymbol{\alpha}$'s for minimal hypergraphs in 9D,10D,...,16D,27D are given in Sec. VI.

Ideally, if we obtained a higher number of 1's for any complete set of data, Eve would be in the line. One can implement the protocol by means of orbital angular momentums which reach over $100 \times 100$ entangled dimensionality [61]. Finally, the huge multiplicity of available hypergraphs enhances the security of QKD, since an attacker may not know which protocol is being used.

### B. Oblivious communication protocol and communication of bounded-dimensional systems protocols

The one-way communication tasks presented in [4], which can make use of any KS set, are communication of a system with bounded dimension, and communication of a system with no dimensional bound, but with some information about the sender's input unrevealed (i.e., oblivious communication). The complete set of vertices (quantum states) in the KS set (in the appropriate dimension) are used as the input alphabet for both the sender's state preparation and receiver's measurement setting. As a result, these quantum communication protocols outperform the best corresponding classical protocols for the same communication tasks.

The protocol we have described is already completely characterized in the research we have cited for lower dimensions, and they demonstrate that any and all KS sets have practical applications. Here we show that the protocol can be straightforwardly extended to any dimension given the necessary computational resources, which scale subexponentially with the dimension.

### C. Generalized Hadamard matrices

Practically all known quantum computation algorithms are based on Fourier transform of which Hadamard ($H$) transform is a special case. Recently, a new $S$ class of $H$ matrices ($S$-$H$) of even order in $\mathbb{C}^n$ has been designed for proving the existence of KS hypergraphs in even $n$D. Our method generates any of these KS hypergraphs (which all turn out to be starlike) and therefore—inversely—the elements of the corresponding $S$-$H$ matrices.

The new $S$ class of $H$ (Hadamard) $S$-$H$ matrices designed in [57] relies on the following theorem.

Theorem (Lisoněk 2019): Suppose that there exists an $S$-$H$ matrix of order $n$ ($n$ even); then there exists a KS hypergraph $k$-$l$ in $\mathbb{C}^n$ such that $k = \binom{n+1}{2}$ and $l = n + 1$

In [57] Lisoněk defines a KS hypergraph in $\mathbb{C}^n$ (Def. 1). By Def. 2.1 he defines an $S$-$H$ matrix and in Def. 2.2 a generalized Hadamard matrix. Via Theorem 3.1 he connects a KS hypergraph and a corresponding $S$-$H$ matrix. The proof provides us with an algorithm for a mutual mapping of their elements in any even dimension. The details are outside the scope of the present paper, and we direct the reader to Ref. [57] for them.

Only the 6D 21-7 KS hypergraph was known to [57], i.e., only one particular $S$-$H$ matrix (apart of the existence of all of them). Our method generates any of these KS hypergraphs together with their coordinatization. Inversely, and that is the core of its application, it gives the elements of the corresponding $S$-$H$ matrices. It also shows that all the corresponding KS hypergraphs are starlike. The latter feature clarifies why $n$ has to be even: one cannot draw a regular star with the Schläfli symbol $\{n+1/\frac{n}{2}\}$ in odd dimensional spaces because $\frac{n}{2}$ has to be an integer. The 8D star 36-9 is given in Sec. IV F. Note the $\{0, \pm 1\}$ coordinatization. The 10D star 55-11 is given in Sec. IV H.

## VI. DISCUSSION

To summarize, we design a feasible unified method of generating quantum contextual sets in higher dimensions as a response to recent calls for high dimensional contextual applications in quantum computation and quantum communication. The need for such a unified method appeared because previous particular attempts were scattered, and the only previously existing unifying method was of exponential complexity and therefore not applicable to 9+ dimensions. This is all presented in detail with references in the Introduction.

Because the complexity of small KS sets does not scale with dimension, the computational resources needed for our method do not generally scale exponentially, depending on

TABLE I. The smallest KS hypergraphs obtained by our methods. A pattern appears to emerge wherein the minimum number of edges per hypergraph of dimension $4n + (0, 1, 2, 3)$ are (9,16,15,14) for integers $n \geqslant 1$, which confirms that the minimum complexity of KS hypergraphs does not grow with dimension; distributions are given in Sec. IV, and MMPH strings and coordinatizations are given in the Appendix.

| Dim. | Smallest critical hypergraphs | $\boldsymbol{\alpha}$ Lovász | No. of non-isom smallest | Smallest master | Vector components |
|---|---|---|---|---|---|
| 4D | 18-9 | 4 | 1 | 24-24 | $\{0, \pm 1\}$ |
| 5D | 29-16 | 7 | 2 | 105-136 | $\{0, \pm 1\}$ |
| 6D | 31-15 | 7 | 1 | 31-16 | $\{0, \pm 1\}$ |
| 7D | 34-14 | 7 | 2 | 47-176 | $\{0, \pm 1\}$ |
| 8D | 34-9 | 4 | 2 | 120-2024 | $\{0, \pm 1\}$ |
| 9D | 47-16 | 7 | 2 | 47-144 | $\{0, \pm 1\}$ |
| 10D | 50-15 | 7 | 66 | 52-141 | $\{0, \pm 1\}$ |
| 11D | 50-14 | 6 | 1603 | 50-38 | $\{0, \pm 1\}$ |
| 12D | 52-9 | 4 | 502 | 52-81 | $\{0, \pm 1\}$ |
| 13D | 63-16 | 7 | 23 | 63-144 | $\{0, \pm 1\}$ |
| 14D | 66-15 | 6 | 17 | 67-157 | $\{0, \pm 1\}$ |
| 15D | 66-14 | 6 | 177 | 66-38 | $\{0, \pm 1\}$ |
| 16D | 68-9 | 4 | 1 | 68-9 | $\{0, \pm 1\}$ |
| ... | ... | ... | ... | ... | ... |
| 27D | 141-16 | 7 | 1 | 141-16 | $\{0, \pm 1\}$ |

the type of new KS sets one is looking for. It is based on dimensional upscaling, i.e., on obtaining higher dimensional contextual sets from previously obtained ones in lower dimensions. Its goal is to generate a manageable-sized master KS hypergraph wherefrom we can obtain a large number of nonisomorphic KS hypergraphs in high dimensional spaces for any possible future application and implementation, e.g., in quantum computation and communication, parity oblivious transfer, genuine random number generation, quantum dimension certification, and relational database theory, three of which are presented in Sec. V. A list of the smallest obtained KS hypergraphs is given in Table I.

Our preliminary testings show that KS hypergraphs in dimensions much higher than 32 can be generated depending on the amount of the CPU time one is ready to sacrifice on supercomputers.

Programs are freely available from our repository [50].

## ACKNOWLEDGMENTS

This work was supported by the Ministry of Science, Education and Youth of Croatia through Center of Excellence CEMS funding and by the European structural and investment funds (ESIF), MSE Grant No. KK.01.1.1.01.0001. Computational support was provided by the Zagreb University Computing Centre.

## APPENDIX: KS MMPH's STRINGS AND COORDINATIZATIONS

Below we provide strings and coordinatizations of KS MMPH's referred to in Sec. IV B–O.

### 1. 4D MMPHs; 9 hyperedges (18-9)

**18-9** 1234,4567,789A,ABCD,DEFG,GHI1,35EC,29BI,68FH. 1=(0,0,0,1), 2=(0,0,1,0), 3=(1,1,0,0), 4=(1,−1,0,0), 5=(0,0,1,1), 6=(1,1,1,−1), 7=(1,1,−1,1), 8=(1,−1,1,1), 9=(1,0,0,−1), A=(0,1,1,0), B=(1,0,0,1), C=(1,−1,1,−1), D=(1,1,−1,−1), E=(1,−1,−1,1), F=(0,1,0,1), G=(1,0,1,0), H=(1,0,−1,0), I=(0,1,0,0)

**24-24** LMNO,HIJK,DEFG,BCFG,9ADE,78EG,56DF,5678,9ABC,68JK,57HI,ACIK,9BHJ,1234,4DGO,3EFN,258M,167L, 19CM,2ABL,3HKO,4IJN,34NO,12LM. 1=(0,0,0,1), 2=(0,0,1,0), 3=(1,1,0,0), 4=(1,−1,0,0), 5=(0,1,0,−1), 6=(1,0,−1,0), 7=(1,0,1,0), 8=(0,1,0,1), 9=(0,1,−1,0), A=(1,0,0,−1), B=(1,0,0,1), C=(0,1,1,0), D=(1,1,1,1), E=(1,−1,−1,1), F=(1,−1,1,−1), G=(1,1,−1,−1), H=(−1,1,1,1), I=(1,1,−1,1), J=(1,1,1,−1), K=(1,−1,1,1), L=(0,1,0,0), M=(1,0,0,0), N=(0,0,1,1), O=(0,0,1,−1)

### 2. 5D MMPH; 16 hyperedges

**29-16** HOINJ,JGSTF,FT4Q8,85679,92LOH,PQRST,KLMNO,CDEIO,ABEGT,34DMO,12BRT,237CO,146AT,279OP,468KT, 5EJOT. 1=(1,−1,1,0,−1), 2=(1,0,−1,0,0), 3=(1,−1,1,1,0), 4=(0,1,1,0,0), 5=(0,0,1,0,0), 6=(1,0,0,0,1), 7=(0,1,0,1,0), 8=(1,0,0,0,−1), 9=(0,1,0,−1,0), A=(1,1,−1,0,−1), B=(1,1,1,0,1), C=(1,1,1,−1,0), D=(1,1,−1,1,0), E=(1,−1,0,0,0), F=(1,−1,1,0,1), G=(0,0,1,0,−1), H=(1,−1,1,−1,0), I=(0,0,1,1,0), J=(1,1,0,0,0), K=(0,1,−1,0,0), L=(1,1,1,1,0), M=(1,0,0,−1,0), N=(1,−1,−1,1,0), O=(0,0,0,0,1), P=(1,0,1,0,0), Q=(1,1,−1,0,1), R=(0,1,0,0,−1), S=(−1,1,1,0,1), T=(0,0,0,1,0)

### 3. 6D MMPHs; 15 to 17 hyperedges

**31-15** A78B91, 194CD3, 3GK65J, J5HLME, EFQSTA, 123456, 2EF9AB, GHI856,NOI756, NP4K56, OPFQR6,OPFLM5, EFRSCU,EFVLDU,EFVMBT. 1=(0,1,0,0,0,0); 2=(0,0,1,0,0,0); G=(1,1,1,1,0,0); H=(1,−1,1,−1,0,0); J=(1,−1,−1,1,0,0); N=(1,−1,−1,−1,0,0); O=(1,1,−1,1,0,0); P=(1,1,1,−1,0,0); E=(1,1,0,0,0,0); F=(1,−1,0,0,0,0); I=(0,1,0,−1,0,0); 7=(1,0,1,0,0,0); 8=(1,0,−1,0,0,0); 3=(1,0,0,−1,0,0); 4=(1,0,0,1,0,0); K=(0,1,−1,0,0,0); 9=(0,0,0,0,1,0); Q=(0,0,1,1,1,1); R=(0,0,1,1,−1,−1); S=(0,0,1,−1,1,−1); V=(0,0,1,−1,1,1); L=(0,0,1,1,−1,1); M=(0,0,1,1,1,−1); 5=(0,0,0,0,1,1); 6=(0,0,0,0,1,−1); C=(0,0,1,0,0,1); D=(0,0,1,0,0,−1); U=(0,0,0,1,1,0); A=(0,0,0,1,0,−1); B=(0,0,0,1,0,1); T=(0,0,1,0,−1,0).

**32-16** 123456, 123789, 12ABCD, 12BE9F, 12ECG5, 12HIG6,12IJ8F,13KL47,23MN47,OPLQ47,ORMS47,PRTBCD, UVTA47, UVTHJD,UWNS47,VWKQ47. 1=(1,0,0,0,0,0); 2=(0,1,0,0,0,0); 3=(0,0,1,0,0,0); O=(1,1,1,1,0,0); P=(1,−1,1,−1,0,0); R=(1,−1,−1,1,0,0); U=(1,−1,−1,−1,0,0); V=(1,−1,1,1,0,0); W=(1,1,1,−1,0,0); T=(1,1,0,0,0,0); A=(0,0,1,−1,0,0); K=(0,1,0,1,0,0); L=(0,1,0,−1,0,0); Q=(1,0,−1,0,0,0); M=(1,0,0,−1,0,0); N=(1,0,0,1,0,0); S=(0,1,−1,0,0,0); 4=(0,0,0,0,1,0); 7=(0,0,0,0,0,1); B=(0,0,1,1,1,1); E=(0,0,1,−1,−1,1); C=(0,0,1,1,−1,−1); H=(0,0,1,−1,−1,−1); I=(0,0,1,1,−1,1); J=(0,0,1,−1,1,1); G=(0,0,1,0,1,0); 5=(0,0,0,1,0,1); 6=(0,0,0,1,0,−1); 8=(0,0,0,1,1,0); 9=(0,0,0,1,−1,0); F=(0,0,1,0,0,−1); D=(0,0,0,0,1,−1).

**33-17** (one of four) 123456,123789,12ABCD,12EFGD,12FH9I,12HGJ5,12BKJ6,12KC8I,13LM47,23NO47,PQMR47, PSNT47,QSUA47,QSUFGD,VWUE47,VXOT47,WXLR47. 1=(1,0,0,0,0,0); 2=(0,1,0,0,0,0); 3=(0,0,1,0,0,0); P=(1,1,1,1,0,0); Q=(1,−1,1,−1,0,0); S=(1,−1,−1,1,0,0); V=(1,−1,−1,−1,0,0); W=(1,−1,1,1,0,0); X=(1,1,1,−1,0,0); U=(1,1,0,0,0,0); A=(0,0,1,1,0,0); E=(0,0,1,−1,0,0); L=(0,1,0,1,0,0); M=(0,1,0,−1,0,0); R=(1,0,−1,0,0,0); N=(1,0,0,−1,0,0); O=(1,0,0,1,0,0); T=(0,1,−1,0,0,0); 4=(0,0,0,0,1,0); 7=(0,0,0,0,0,1); F=(0,0,1,1,1,1); H=(0,0,1,−1,−1,1); G=(0,0,1,1,−1,−1); B=(0,0,1,−1,−1,−1); K=(0,0,1,1,−1,1); C=(0,0,1,−1,1,1); J=(0,0,1,0,1,0); 5=(0,0,0,1,0,1); 6=(0,0,0,1,0,−1); 8=(0,0,0,1,1,0); 9=(0,0,0,1,−1,0); I=(0,0,1,0,0,−1); D=(0,0,0,0,1,−1).

#### 4. 7D MMPH; 14 hyperedges

**34-14** 1234567, 189A5BC, 189DE7F, 189GHIJ, 189KHBL, 2MNDOIP, 2MNEOCL, 2MNGK6F, QRNSAJP, QT4U567, RTV9567, WXMS567, WYV8AJP, XY3U567. 1=(0,0,0,1,0,0,0); 2=(0,0,1,0,0,0,0); 3=(1,−1,0,0,0,0,0); 4=(1,1,0,0,0,0,0); 5=(0,0,0,0,0,0,1); 6=(0,0,0,0,1,1,0); 7=(0,0,0,0,1,−1,0); 8=(0,1,−1,0,0,0,0); 9=(0,1,1,0,0,0,0); A=(0,0,0,0,1,0,0); B=(1,0,0,0,0,−1,0); C=(1,0,0,0,0,1,0); D=(1,0,0,0,1,1,−1); E=(−1,0,0,0,1,1,1); F=(1,0,0,0,0,0,1); G=(1,0,0,0,1,−1,−1); H=(1,0,0,0,1,1,1); I=(1,0,0,0,−1,0,0); J=(0,0,0,0,0,1,−1); K=(1,0,0,0,−1,1,−1); L=(0,0,0,0,1,0,−1); M=(0,1,0,1,0,0,0); N=(0,1,0,−1,0,0,0); O=(1,0,0,0,1,−1,1); P=(0,0,0,0,0,1,1); Q=(−1,1,1,1,0,0,0); R=(1,1,−1,1,0,0,0); S=(1,0,1,0,0,0,0); T=(1,−1,1,1,0,0,0); U=(0,0,1,−1,0,0,0); V=(1,0,0,−1,0,0,0); W=(1,−1,−1,1,0,0,0); X=(1,1,−1,−1,0,0,0); Y=(1,1,1,1,0,0,0).

#### 5. 8D MMPH; 9 hyperedges

**34-9** 12345678,9ABCDEFG,HIJKLMFG,NOPQME78,RSTULD56,VWXUKC46,XTPQJB36,YVWOIA28,YRSNH918. 1=(0,0,1,0,0,0,0,0), 2=(1,0,0,0,0,0,0,0), 3=(0,0,0,0,0,1,0,0), 4=(0,0,0,0,1,0,0,0), 5=(0,0,0,1,0,0,0,0), 6=(0,0,0,0,0,0,0,1), 7=(0,1,0,0,0,0,0,0), 8=(0,0,0,0,0,0,1,0), 9=(1,1,0,−1,0,0,0,−1), A=(0,1,−1,0,−1,0,0,1), B=(0,1,0,1,1,0,−1,0), C=(1,0,0,1,0,−1,1,0), D=(0,0,−1,0,1,1,1,0), E=(1,0,1,0,0,1,0,1), F=(−1,1,1,0,0,0,1,0), G=(0,0,0,1,−1,1,0,−1), H=(1,1,0,1,0,0,0,1), I=(0,1,−1,0,1,0,0,−1), J=(0,−1,0,1,1,0,1,0), K=(1,0,0,−1,0,1,1,0), L=(0,0,1,0,1,1,−1,0), M=(1,0,1,0,0,−1,0,−1), N=(0,0,0,1,1,1,0,−1), O=(0,0,0,1,1,−1,0,1), P=(0,0,0,1,−1,0,0,0), Q=(1,0,−1,0,0,0,0,0), R=(0,0,0,0,1,−1,0,0), S=(1,−1,0,0,0,0,0,0), T=(1,1,1,0,0,0,1,0), U=(1,1,−1,0,0,0,−1,0), V=(0,0,0,1,0,1,0,0), W=(0,1,1,0,0,0,0,0), X=(1,−1,1,0,0,0,−1,0), Y=(0,0,0,−1,1,1,0,1)

**36-9** 12345678,89ABCDEF,FGHI4JKL,L7MNBOPQ,QERSI3TU,UK6VNAWX,XPDYSH2Z,ZTJ5VM9a,aWOCYRG1. 1=(0,0,0,0,0,0,0,1), 2=(0,0,0,0,0,0,1,0), 3=(0,0,0,0,0,1,0,0), 4=(0,0,0,0,1,0,0,0), 5=(0,0,1,1,0,0,0,0), 6=(0,0,1,−1,0,0,0,0), 7=(1,1,0,0,0,0,0,0), 8=(1,−1,0,0,0,0,0,0), 9=(1,1,0,0,0,0,−1,1), A=(0,0,1,1,1,−1,0,0), B=(0,0,0,0,0,0,1,1), C=(0,0,1,−1,1,1,0,0), D=(0,0,0,1,0,1,0,0), E=(0,0,1,0,−1,0,0,0), F=(1,1,0,0,0,0,1,−1), G=(0,0,1,0,0,−1,0,0), H=(1,0,0,0,0,0,0,1), I=(0,0,0,1,0,0,0,0), J=(1,−1,0,0,0,0,−1,−1), K=(0,1,0,0,0,0,−1,0), L=(0,0,1,0,0,1,0,0), M=(0,0,1,−1,1,−1,0,0), N=(1,−1,0,0,0,0,−1,1), O=(0,0,0,1,1,0,0,0), P=(0,0,1,1,−1,−1,0,0), Q=(1,−1,0,0,0,0,1,−1), R=(1,0,0,0,0,0,−1,0), S=(0,0,1,0,1,0,0,0), T=(0,1,0,0,0,0,0,−1), U=(1,1,0,0,0,0,1,1), V=(0,0,0,0,1,1,0,0), W=(0,0,1,1,−1,1,0,0), X=(1,0,0,0,0,0,0,−1), Y=(0,1,0,0,0,0,0,0), Z=(0,0,1,−1,−1,1,0,0), a=(1,0,0,0,0,0,1,0)

#### 6. 9D MMPH; 16 hyperedges

**47-16** 234567891,1BOPERSNQ,QdgXJDaFY,YaFZIbc32,12ABC5DEF,13GHIJ7KF,1A4567LMN,1G4567STU,1HOV6KWLU, 1XOV6KRT9,1XIJ7KWM8,deHfOV6KF,eghA4567F,ijhBOPEQF,ikZXPVblF,jkGfC4lcF. 1=(1,0,0,0,0,0,0,0,0); 2=(0,1,0,0,0,0,0,0,0); 3=(0,0,1,0,0,0,0,0,0); d=(1,1,1,1,0,0,0,0,0); e=(1,−1,1,−1,0,0,0,0,0); g=(1,−1,−1,1,0,0,0,0,0); i=(1,−1,−1,−1,0,0,0,0,0); j=(1,−1,1,1,0,0,0,0,0); k=(1,1,1,−1,0,0,0,0,0); h=(1,1,0,0,0,0,0,0,0); A=(0,0,1,1,0,0,0,0,0); B=(0,0,1,−1,0,0,0,0,0); G=(0,1,0,1,0,0,0,0,0); H=(0,1,0,−1,0,0,0,0,0); f=(1,0,−1,0,0,0,0,0,0); Y=(1,0,0,−1,0,0,0,0,0); Z=(1,0,0,1,0,0,0,0,0); X=(0,1,−1,0,0,0,0,0,0); C=(0,0,0,0,1,0,0,0,0); 4=(0,0,0,0,0,1,0,0,0); 5=(0,0,0,0,0,0,1,0,0); O=(0,0,0,0,1,1,1,1,0); P=(0,0,0,0,1,−1,1,−1,0); V=(0,0,0,0,1,−1,−1,1,0); I=(0,0,0,0,1,−1,−1,−1,0); a=(0,0,0,0,1,−1,1,1,0); J=(0,0,0,0,1,1,1,−1,0); b=(0,0,0,0,1,1,0,0,0); l=(0,0,0,0,0,0,1,1,0); c=(0,0,0,0,0,0,1,−1,0); D=(0,0,0,0,0,1,0,1,0); E=(0,0,0,0,0,1,0,−1,0); Q=(0,0,0,0,1,0,−1,0,0); 6=(0,0,0,0,1,0,0,−1,0); 7=(0,0,0,0,1,0,0,1,0); K=(0,0,0,0,0,1,−1,0,0); F=(0,0,0,0,0,0,0,0,1); W=(0,1,1,1,0,0,0,0,1); L=(0,1,−1,1,0,0,0,0,−1); M=(0,1,1,−1,0,0,0,0,−1); R=(0,1,1,1,0,0,0,0,−1); S=(0,1,−1,−1,0,0,0,0,−1); T=(0,1,1,−1,0,0,0,0,1); N=(0,1,0,0,0,0,0,0,1); U=(0,0,1,0,0,0,0,0,−1); 8=(0,0,0,1,0,0,0,0,−1); 9=(0,0,0,1,0,0,0,0,1)

#### 7. 10D MMPH; 11 and 15 hyperedges

Star-like **55-11** 123456789A,ABCDEFGHIJ,JKLMN5OPQR,R9STUEVWXY,YIZabN4cde,eQ8fgUDhij,jXHklbM3mn, ndP7ogTCpq,qiWGrlaL2s,smcO6ofSBt,tphVFrkZK1.

**50-15** 2ajkBCJST1,1TEOPRbmlD,Dl89FghnVU,UV46IceoqG,GHLNXZajk2,1DEJXYdeij,1DELMVWajk,2GHJXYfhpq, 2GHMNRSblm,2GHOQYZblm,2GHPQUWajk,45EFUVabop,56ABUVcdmn,78ACUVfgik,79HIJSTblm. 1=(1,0,0,0,0,0,0,0,0,0); 2=(0,1,0,0,0,0,0,0,0,0); 4=(1,1,1,1,0,0,0,0,0,0); 5=(1,−1,1,−1,0,0,0,0,0,0); 6=(1,−1,−1,1,0,0,0,0,0,0); 7=(1,−1,−1,−1,0,0,0,0,0,0); 8=(1,−1,1,1,0,0,0,0,0,0); 9=(1,1,1,−1,0,0,0,0,0,0); A=(1,1,0,0,0,0,0,0,0,0); B=(0,0,1,1,0,0,0,0,0,0); C=(0,0,1,−1,0,0,0,0,0,0); D=(0,1,0,1,0,0,0,0,0,0); E=(0,1,0,−1,0,0,0,0,0,0); F=(1,0,−1,0,0,0,0,0,0,0); G=(1,0,0,−1,0,0,0,0,0,0); H=(1,0,0,1,0,0,0,0,0,0); I=(0,1,−1,0,0,0,0,0,0,0); J=(0,0,0,0,1,0,0,0,0,0); L=(0,0,1,0,1,1,1,0,0,0); M=(0,0,1,0,−1,1,−1,0,0,0); N=(0,0,1,0,−1,−1,1,0,0,0); O=(0,0,1,0,−1,−1,−1,0,0,0); P=(0,0,1,0,−1,1,1,0,0,0); Q=(0,0,1,0,1,1,−1,0,0,0); R=(0,0,1,0,1,0,0,0,0,0); S=(0,0,0,0,0,1,1,0,0,0); T=(0,0,0,0,0,1,−1,0,0,0); U=(0,0,0,0,1,0,1,0,0,0); V=(0,0,0,0,1,0,−1,0,0,0); W=(0,0,1,0,0,−1,0,0,0,0); X=(0,0,1,0,0,0,−1,0,0,0); Y=(0,0,1,0,0,0,1,0,0,0); Z=(0,0,0,0,1,−1,0,0,0,0); a=(0,0,0,0,0,0,0,1,0,0); b=(0,0,0,0,0,0,0,0,1,0); c=(0,0,0,0,0,1,0,1,1,1); d=(0,0,0,0,0,1,0,−1,1,−1);

e=(0,0,0,0,0,1,0,−1,−1,1); f=(0,0,0,0,0,1,0,−1,−1,−1); g=(0,0,0,0,0,1,0,−1,1,1); h=(0,0,0,0,0,1,0,1,1,−1);
i=(0,0,0,0,0,1,0,1,0,0); j=(0,0,0,0,0,0,0,0,1,1); k=(0,0,0,0,0,0,0,0,1,−1); l=(0,0,0,0,0,0,0,1,0,1);
m=(0,0,0,0,0,0,0,1,0,−1); n=(0,0,0,0,0,1,0,0,−1,0); o=(0,0,0,0,0,1,0,0,0,−1); p=(0,0,0,0,0,1,0,0,0,1);
q=(0,0,0,0,0,0,0,1,−1,0)

### 8. 11D MMPH; 14 hyperedges

**50-14** 567E234CFD1,1DGHJKLMANI,IYlmnUk9QXe,ecdfgOW67E5,123456789AB,1GHIJKLOPQR,27STUVKL8FW,
27STUVKL9QX,347YZabMDAN,567cdefMCXR,567cdefgPBN,cdhijkJV8FW,fZoHTjmn8FW,abGShilo8FW.
1=(0,0,1,1,1,1,0,0,0,0,0); 2=(0,0,1,−1,1,−1,0,0,0,0,0); 3=(0,0,0,1,0,−1,0,0,0,0,0); 4=(0,0,1,0,−1,0,0,0,0,0,0);
5=(0,1,0,0,0,0,0,0,0,0,0); 6=(1,0,0,0,0,0,0,0,0,0,0); 7=(0,0,0,0,0,0,1,0,0,0,0); 8=(0,0,0,1,0,0,0,0,0,0,0);
9=(0,0,1,0,0,0,0,0,0,0,0); A=(0,0,0,0,0,1,0,0,0,0,0); B=(0,0,0,0,1,0,0,0,0,0,0); C=(1,−1,1,0,1,0,0,0,0,0,0);
D=(1,1,0,1,0,1,0,0,0,0,0); E=(1,1,0,−1,0,−1,0,0,0,0,0); F=(−1,1,1,0,1,0,0,0,0,0,0); G=(0,1,−1,1,0,0,1,0,0,0,0);
H=(1,0,1,1,0,0,0,−1,0,0,0); I=(1,0,0,0,1,1,0,1,0,0,0); J=(0,1,0,0,−1,1,−1,0,0,0,0); K=(0,0,1,0,−1,0,1,1,0,0,0);
L=(0,0,0,1,0,−1,−1,1,0,0,0); M=(1,0,1,0,0,−1,1,0,0,0,0); N=(0,−1,1,0,0,1,0,1,0,0,0); O=(−1,1,0,0,0,0,1,1,0,0,0);
P=(1,0,−1,−1,0,0,0,1,0,0,0); Q=(0,1,1,−1,0,0,−1,0,0,0,0); R=(1,0,0,1,−1,0,−1,0,0,0,0); S=(0,1,0,1,1,0,0,1,0,0,0);
T=(1,1,0,0,0,0,1,−1,0,0,0); U=(0,1,0,0,1,−1,−1,0,0,0,0); V=(1,0,0,0,−1,−1,0,1,0,0,0); W=(1,1,0,−1,0,1,0,0,0,0,0);
X=(1,−1,−1,0,1,0,0,0,0,0,0); Y=(0,0,0,0,0,0,0,0,1,0,0); Z=(0,0,0,0,0,0,0,0,0,1,0); a=(0,0,0,0,0,0,0,1,1,1,1);
b=(0,0,0,0,0,0,0,1,−1,1,−1); c=(0,0,0,0,0,0,0,1,−1,−1,1); d=(0,0,0,0,0,0,0,1,−1,−1,−1); e=(0,0,0,0,0,0,0,1,−1,1,1);
f=(0,0,0,0,0,0,0,1,1,1,−1); g=(0,0,0,0,0,0,0,1,1,0,0); h=(0,0,0,0,0,0,0,0,0,1,1); i=(0,0,0,0,0,0,0,0,0,1,−1);
j=(0,0,0,0,0,0,0,0,1,0,1); k=(0,0,0,0,0,0,0,0,1,0,−1); l=(0,0,0,0,0,0,0,1,0,−1,0); m=(0,0,0,0,0,0,0,1,0,0,−1);
n=(0,0,0,0,0,0,0,1,0,0,1); o=(0,0,0,0,0,0,0,0,1,−1,0)

### 9. 12D MMPH; 9 hyperedges

**52-9** UVX34STYR8W7,78W56bcdQaeZ,ZaehiGPjJgkf,fgkDMlpqLVXU,123456789ABC,17DEFGHIJBKL,28MNOPHIQAKR,
bSlmnFOijdoC,cTpENhmn9qYo. 1=(0,0,1,1,1,1,0,0,0,0,0,0); 2=(0,0,1,−1,1,−1,0,0,0,0,0,0); 3=(0,0,0,1,0,−1,0,0,0,0,0,0);
4=(0,0,1,0,−1,0,0,0,0,0,0,0); 5=(0,1,0,0,0,0,0,0,0,0,0,0); 6=(1,0,0,0,0,0,0,0,0,0,0,0); 7=(0,0,0,0,0,0,0,1,0,0,0,0);
8=(0,0,0,0,0,0,1,0,0,0,0,0); Z=(0,0,0,1,0,0,0,0,0,0,0,0); a=(0,0,1,0,0,0,0,0,0,0,0,0); b=(0,0,0,0,0,1,0,0,0,0,0,0);
c=(0,0,0,0,1,0,0,0,0,0,0,0); S=(1,−1,1,0,1,0,0,0,0,0,0,0); T=(1,1,0,1,0,1,0,0,0,0,0,0); U=(1,1,0,−1,0,−1,0,0,0,0,0,0);
V=(−1,1,1,0,1,0,0,0,0,0,0,0); D=(0,1,−1,1,0,0,1,0,0,0,0,0); M=(1,0,1,1,0,0,0,−1,0,0,0,0); f=(1,0,0,0,1,1,0,1,0,0,0,0);
g=(0,1,0,0,−1,1,−1,0,0,0,0,0); l=(0,0,1,0,−1,0,1,1,0,0,0,0); p=(0,0,0,1,0,−1,−1,1,0,0,0,0); E=(1,0,1,0,0,−1,1,0,0,0,0,0);
N=(0,−1,1,0,0,1,0,1,0,0,0,0); h=(−1,1,0,0,0,0,1,1,0,0,0,0); m=(1,0,−1,−1,0,0,0,1,0,0,0,0); n=(0,1,1,−1,0,0,−1,0,0,0,0,0);
F=(1,0,0,1,−1,0,−1,0,0,0,0,0); O=(0,1,0,1,1,0,0,1,0,0,0,0); i=(1,1,0,0,0,0,1,−1,0,0,0,0); G=(0,1,0,0,1,−1,−1,0,0,0,0,0);
P=(1,0,0,0,−1,−1,0,1,0,0,0,0); H=(1,1,0,−1,0,1,0,0,0,0,0,0); I=(1,−1,−1,0,1,0,0,0,0,0,0,0); j=(0,0,0,0,0,0,0,0,1,0,0,0);
d=(0,0,0,0,0,0,0,0,0,1,0,0); J=(0,0,0,0,0,0,0,0,0,1,1,0); k=(0,0,0,0,0,0,0,0,0,1,−1,0); W=(0,0,0,0,0,0,0,0,1,0,1,0);
Q=(0,0,0,0,0,0,0,0,1,0,−1,0); 9=(0,0,0,0,0,0,0,0,1,−1,0,0); e=(0,0,0,0,0,0,0,0,0,0,0,1); A=(0,0,0,0,0,0,0,0,1,1,1,1);
B=(0,0,0,0,0,0,0,0,1,1,−1,−1); K=(0,0,0,0,0,0,0,0,1,−1,1,−1); X=(0,0,0,0,0,0,0,0,1,−1,−1,−1); q=(0,0,0,0,0,0,0,0,1,1,1,−1);
Y=(0,0,0,0,0,0,0,0,1,1,−1,1); L=(0,0,0,0,0,0,0,0,1,0,0,1); o=(0,0,0,0,0,0,0,0,0,0,1,1); C=(0,0,0,0,0,0,0,0,0,0,1,−1);
R=(0,0,0,0,0,0,0,0,0,1,0,−1)

### 10. 13D MMPHs; 16 hyperedges

**63-16** 123456789ABCD,123456789EFGH,12345678IJKLM,17NOPQRSTUVWM,28XYZaRS9EbBM,3478cdef9bghH,
3478cdef9WhiC,3478cdefIjVkM,5678lmno9LpGq,5678lmnoATrBM,lmstuvQa9WpFD,lmstuvQaEKwxM,
ncyz!PZv9AkLM,od''OYuz!9kgiq,od''OYuz!bUjxM,efNXsty''rJwWM. 1=(0,0,1,1,1,1,0,0,0,0,0,0,0);
2=(0,0,1,−1,1,−1,0,0,0,0,0,0,0); 3=(0,0,0,1,0,−1,0,0,0,0,0,0,0); 4=(0,0,1,0,−1,0,0,0,0,0,0,0,0); 5=(0,1,0,0,0,0,0,0,0,0,0,0,0);
6=(1,0,0,0,0,0,0,0,0,0,0,0,0); 7=(0,0,0,0,0,0,0,1,0,0,0,0,0); 8=(0,0,0,0,0,0,1,0,0,0,0,0,0); l=(0,0,0,1,0,0,0,0,0,0,0,0,0);
m=(0,0,1,0,0,0,0,0,0,0,0,0,0); n=(0,0,0,0,0,1,0,0,0,0,0,0,0); o=(0,0,0,0,1,0,0,0,0,0,0,0,0); c=(1,−1,1,0,1,0,0,0,0,0,0,0,0);
d=(1,1,0,1,0,1,0,0,0,0,0,0,0); e=(1,1,0,−1,0,−1,0,0,0,0,0,0,0); f=(−1,1,1,0,1,0,0,0,0,0,0,0,0); N=(0,1,−1,1,0,0,1,0,0,0,0,0,0);
X=(1,0,1,1,0,0,0,−1,0,0,0,0,0); s=(1,0,0,0,1,1,0,1,0,0,0,0,0); t=(0,1,0,0,−1,1,−1,0,0,0,0,0,0); y=(0,0,1,0,−1,0,1,1,0,0,0,0,0);
''=(0,0,0,1,0,−1,−1,1,0,0,0,0,0); O=(1,0,1,0,0,−1,1,0,0,0,0,0,0); Y=(0,−1,1,0,0,1,0,1,0,0,0,0,0);
u=(−1,1,0,0,0,0,1,1,0,0,0,0,0); z=(1,0,−1,−1,0,0,0,1,0,0,0,0,0); !=(0,1,1,−1,0,0,−1,0,0,0,0,0,0);
P=(1,0,0,1,−1,0,−1,0,0,0,0,0,0); Z=(0,1,0,1,1,0,0,1,0,0,0,0,0); v=(1,1,0,0,0,0,1,−1,0,0,0,0,0); Q=(0,1,0,0,1,−1,−1,0,0,0,0,0,0);
a=(1,0,0,0,−1,−1,0,1,0,0,0,0,0); R=(1,1,0,−1,0,1,0,0,0,0,0,0,0); S=(1,−1,−1,0,1,0,0,0,0,0,0,0,0);
9=(0,0,0,0,0,0,0,0,1,0,0,0,0); A=(0,0,0,0,0,0,0,0,0,1,0,0,0); E=(0,0,0,0,0,0,0,0,0,1,1,0,0); b=(0,0,0,0,0,0,0,0,0,1,−1,0,0);
T=(0,0,0,0,0,0,0,0,1,0,1,0,0); r=(0,0,0,0,0,0,0,0,1,0,−1,0,0); I=(0,0,0,0,0,0,0,0,1,−1,0,0,0); B=(0,0,0,0,0,0,0,0,0,0,0,1,0);
J=(0,0,0,0,0,0,0,0,1,1,1,1,0); K=(0,0,0,0,0,0,0,0,1,1,−1,−1,0); w=(0,0,0,0,0,0,0,0,1,−1,1,−1,0);
U=(0,0,0,0,0,0,0,0,1,−1,−1,−1,0); j=(0,0,0,0,0,0,0,0,1,1,1,−1,0);V=(0,0,0,0,0,0,0,0,1,1,−1,1,0); x=(0,0,0,0,0,0,0,0,1,0,0,1,0);
k=(0,0,0,0,0,0,0,0,0,0,1,1,0); L=(0,0,0,0,0,0,0,0,0,0,1,−1,0); W=(0,0,0,0,0,0,0,0,0,1,0,−1,0); M=(0,0,0,0,0,0,0,0,0,0,0,0,1);
p=(0,0,0,0,0,0,0,0,0,1,1,1,1); F=(0,0,0,0,0,0,0,0,0,1,−1,1,−1); G=(0,0,0,0,0,0,0,0,0,1,−1,−1,1); g=(0,0,0,0,0,0,0,0,0,1,1,−1,1);

h=(0,0,0,0,0,0,0,0,0,1,1,1,−1); i=(0,0,0,0,0,0,0,0,0,1,−1,1,1); H=(0,0,0,0,0,0,0,0,0,0,0,0,1,1); C=(0,0,0,0,0,0,0,0,0,0,0,1,0,1); D=(0,0,0,0,0,0,0,0,0,0,0,1,0,−1); q=(0,0,0,0,0,0,0,0,0,0,1,0,0,−1)

### 11. 14D MMPH; 15 hyperedges

**66-15** `347E25689ABCD1,1IKLMNOPFQRSTJ,Jr$Vx!''s%abcke,ekdfghijlm47E3,12345679ABFGDH,27UVWXMNYZTabc,347defg9ABPHnm,347defgFijkGDH,567opqrOBPFsZt,567opqr9ABaunl,567opqrijbkCGu,opvwxyLX9iQYab,qdz!''KWyAt#Sab,fgIUvwz$Oh89AB,fgIUvwz$#R%jab.` 1=(0,0,1,1,1,1,0,0,0,0,0,0,0,0);
2=(0,0,1,−1,1,−1,0,0,0,0,0,0,0,0); 3=(0,0,0,1,0,−1,0,0,0,0,0,0,0,0); 4=(0,0,1,0,−1,0,0,0,0,0,0,0,0,0);
5=(0,1,0,0,0,0,0,0,0,0,0,0,0,0); 6=(1,0,0,0,0,0,0,0,0,0,0,0,0,0); 7=(0,0,0,0,0,0,1,0,0,0,0,0,0,0);
o=(0,0,0,1,0,0,0,0,0,0,0,0,0,0); p=(0,0,1,0,0,0,0,0,0,0,0,0,0,0); q=(0,0,0,0,0,1,0,0,0,0,0,0,0,0);
r=(0,0,0,0,1,0,0,0,0,0,0,0,0,0); d=(1,−1,1,0,1,0,0,0,0,0,0,0,0,0); e=(1,1,0,1,0,1,0,0,0,0,0,0,0,0);
f=(1,1,0,−1,0,−1,0,0,0,0,0,0,0,0); g=(−1,1,1,0,1,0,0,0,0,0,0,0,0,0); I=(0,1,−1,1,0,0,1,0,0,0,0,0,0,0);
U=(1,0,1,1,0,0,0,−1,0,0,0,0,0,0); v=(1,0,0,0,1,1,0,1,0,0,0,0,0,0); w=(0,1,0,0,−1,1,−1,0,0,0,0,0,0,0);
z=(0,0,1,0,−1,0,1,1,0,0,0,0,0,0); $=(0,0,0,1,0,−1,−1,1,0,0,0,0,0,0); J=(1,0,1,0,0,−1,1,0,0,0,0,0,0,0);
V=(0,−1,1,0,0,1,0,1,0,0,0,0,0,0); x=(−1,1,0,0,0,0,1,1,0,0,0,0,0,0); !=(1,0,−1,−1,0,0,0,1,0,0,0,0,0,0);
''=(0,1,1,−1,0,0,−1,0,0,0,0,0,0,0); K=(1,0,0,1,−1,0,−1,0,0,0,0,0,0,0); W=(0,1,0,1,1,0,0,1,0,0,0,0,0,0);
y=(1,1,0,0,0,0,1,−1,0,0,0,0,0,0); L=(0,1,0,0,1,−1,−1,0,0,0,0,0,0,0); X=(1,0,0,0,−1,−1,0,1,0,0,0,0,0,0);
M=(1,1,0,−1,0,1,0,0,0,0,0,0,0,0); N=(1,−1,−1,0,1,0,0,0,0,0,0,0,0,0); O=(0,0,0,0,0,0,0,0,0,0,1,0,0,0);
h=(0,0,0,0,0,0,0,0,1,−1,0,0,0,0); 8=(0,0,0,0,0,0,0,0,1,1,0,0,0,0); 9=(0,0,0,0,0,0,0,0,0,0,0,0,0,1);
A=(0,0,0,0,0,0,0,0,0,0,0,1,1,0); B=(0,0,0,0,0,0,0,0,0,0,0,1,−1,0); P=(0,0,0,0,0,0,0,0,1,0,−1,0,0,0,0);
F=(0,0,0,0,0,0,0,0,1,0,1,0,0,0,0); i=(0,0,0,0,0,0,0,0,0,0,0,1,0,0); Q=(0,0,0,0,0,0,0,0,0,1,0,0,0,−1,0);
Y=(0,0,0,0,0,0,0,0,1,0,0,0,1,0); s=(0,0,0,0,0,0,0,0,1,0,0,1,1,−1); Z=(0,0,0,0,0,0,0,0,1,0,0,−1,−1,−1);
t=(0,0,0,0,0,0,0,0,1,0,0,0,0,1); #=(0,0,0,0,0,0,0,0,1,0,0,1,−1,−1); R=(0,0,0,0,0,0,0,0,0,1,0,0,1,1,1);
%=(0,0,0,0,0,0,0,0,1,0,0,−1,0,0); j=(0,0,0,0,0,0,0,0,0,0,0,0,1,−1); S=(0,0,0,0,0,0,0,0,1,0,0,−1,1,−1);
T=(0,0,0,0,0,0,0,0,0,0,0,1,0,−1); a=(0,0,0,0,0,0,0,0,0,1,1,0,0,0); b=(0,0,0,0,0,0,0,0,0,0,1,−1,0,0,0);
c=(0,0,0,0,0,0,0,0,1,0,0,1,−1,1); k=(0,0,0,0,0,0,0,0,0,0,0,0,1,1); C=(0,0,0,0,0,0,0,1,−1,1,1,0,0,0);
G=(0,0,0,0,0,0,0,1,−1,−1,−1,0,0,0); u=(0,0,0,0,0,0,0,1,1,0,0,0,0,0); D=(0,0,0,0,0,0,0,1,1,−1,1,0,0,0);
E=(0,0,0,0,0,0,0,1,0,0,−1,0,0,0); H=(0,0,0,0,0,0,0,0,0,1,0,−1,0,0,0); n=(0,0,0,0,0,0,0,1,−1,1,−1,0,0,0);
l=(0,0,0,0,0,0,0,1,−1,−1,1,0,0,0); m=(0,0,0,0,0,0,0,1,1,1,1,0,0,0)

### 12. 15D MMPH; 14 hyperedges

**66-14** `fg1256v!''#47pq3,347pqDEFlmnorsC,CAKPQRSYZabcdeT,Tcde89IJOUVhigf,1234567Ydnty''$%,1GMRUWXYdnty''$%,27HNSVWXcdefghi,347CDEFhjrwxyz!,347CDEFikstuvwx,56789ABZeouz#$%,56789ABabdejklm,89IJOTUVYZabcde,BDLMNOPQYZabcde,EFGHIJKLYZabcde.` 1=(0,0,1,1,1,1.0,0,0,0,0,0,0,0,0);
2=(0,0,1,−1,1,−1.0,0,0,0,0,0,0,0,0); 3=(0,0,0,1,0,−1.0,0,0,0,0,0,0,0,0); 4=(0,0,1,0,−1,0.0,0,0,0,0,0,0,0,0);
5=(0,1,0,0,0,0.0,0,0,0,0,0,0,0,0); 6=(1,0,0,0,0,0.0,0,0,0,0,0,0,0,0); 7=(0,0,0,0,0,0.1,0,0,0,0,0,0,0,0);
8=(0,0,0,1,0,0.0,0,0,0,0,0,0,0,0); 9=(0,0,1,0,0,0.0,0,0,0,0,0,0,0,0); A=(0,0,0,0,0,1.0,0,0,0,0,0,0,0,0);
B=(0,0,0,0,1,0.0,0,0,0,0,0,0,0,0); C=(1,−1,1,0,1,0.0,0,0,0,0,0,0,0,0); D=(1,1,0,1,0,1.0,0,0,0,0,0,0,0,0);
E=(1,1,0,−1,0,−1.0,0,0,0,0,0,0,0,0); F=(−1,1,1,0,1,0.0,0,0,0,0,0,0,0,0); G=(0,1,−1,1,0,0.1,0,0,0,0,0,0,0,0);
H=(1,0,1,1,0,0.0,−1,0,0,0,0,0,0,0); I=(1,0,0,0,1,1.0,1,0,0,0,0,0,0,0); J=(0,1,0,0,−1,1.−1,0,0,0,0,0,0,0,0);
K=(0,0,1,0,−1,0.1,1,0,0,0,0,0,0,0); L=(0,0,0,1,0,−1.−1,1,0,0,0,0,0,0,0); M=(1,0,1,0,0,−1.1,0,0,0,0,0,0,0,0);
N=(0,−1,1,0,0,1.0,1,0,0,0,0,0,0,0); O=(−1,1,0,0,0,0.1,1,0,0,0,0,0,0,0); P=(1,0,−1,−1,0,0.0,1,0,0,0,0,0,0,0);
Q=(0,1,1,−1,0,0.−1,0,0,0,0,0,0,0,0); R=(1,0,0,1,−1,0.−1,0,0,0,0,0,0,0,0); S=(0,1,0,1,1,0.0,1,0,0,0,0,0,0,0);
T=(1,1,0,0,0,0.1,−1,0,0,0,0,0,0,0); U=(0,1,0,0,1,−1.−1,0,0,0,0,0,0,0,0); V=(1,0,0,0,−1,−1.0,1,0,0,0,0,0,0,0);
W=(1,1,0,−1,0,1.0,0,0,0,0,0,0,0,0); X=(1,−1,−1,0,1,0.0,0,0,0,0,0,0,0,0); Y=(0,0,0,0,0,0.0,0,0,1,1,1,1,0,0);
Z=(0,0,0,0,0,0.0,0,0,1,−1,1,−1,0,0); a=(0,0,0,0,0,0.0,0,0,0,1,0,−1,0,0); b=(0,0,0,0,0,0.0,0,0,1,0,−1,0,0,0);
c=(0,0,0,0,0,0.0,0,1,0,0,0,0,0,0); d=(0,0,0,0,0,0.0,0,0,0,0,0,0,0,1); e=(0,0,0,0,0,0.0,0,0,0,0,0,0,1,0);
f=(0,0,0,0,0,0.0,0,0,0,1,0,0,0,0); g=(0,0,0,0,0,0.0,0,0,1,0,0,0,0,0); h=(0,0,0,0,0,0.0,0,0,0,0,0,1,0,0);
i=(0,0,0,0,0,0.0,0,0,0,0,1,0,0,0); j=(0,0,0,0,0,0.0,1,−1,1,0,1,0,0,0); k=(0,0,0,0,0,0.0,1,1,0,1,0,1,0,0);
l=(0,0,0,0,0,0.0,1,1,0,−1,0,−1,0,0); m=(0,0,0,0,0,0.0,1,−1,−1,0,−1,0,0,0); n=(0,0,0,0,0,0.0,0,1,−1,1,0,0,1,0);
o=(0,0,0,0,0,0.0,1,0,1,1,0,0,0,−1); p=(0,0,0,0,0,0.0,1,0,0,0,1,1,0,1); q=(0,0,0,0,0,0.0,0,1,0,0,−1,1,−1,0);
r=(0,0,0,0,0,0.0,0,0,1,0,−1,0,1,1); s=(0,0,0,0,0,0.0,0,0,0,1,0,−1,−1,1); t=(0,0,0,0,0,0.0,1,0,1,0,0,−1,1,0);
u=(0,0,0,0,0,0.0,0,1,−1,0,0,−1,0,−1); v=(0,0,0,0,0,0.0,1,−1,0,0,0,0,−1,−1); w=(0,0,0,0,0,0.0,1,0,−1,−1,0,0,0,1);
x=(0,0,0,0,0,0.0,0,1,1,−1,0,0,−1,0); y=(0,0,0,0,0,0.0,1,0,0,1,−1,0,−1,0); z=(0,0,0,0,0,0.0,0,1,0,1,1,0,0,1);
!=(0,0,0,0,0,0.0,1,1,0,0,0,0,1,−1); ''=(0,0,0,0,0,0.0,0,1,0,0,1,−1,−1,0); #=(0,0,0,0,0,0.0,1,0,0,0,−1,−1,0,1);
$=(0,0,0,0,0,0.0,1,1,0,−1,0,1,0,0); %=(0,0,0,0,0,0.0,1,−1,−1,0,1,0,0,0).

### 13. 16D MMPH; 9 hyperedges

**70-9** 28)3456Zaop#%7v1,17vHNSVehjqsyYwX,XYwIOTWcklnz''8)2,3478DEFGbijnu#$&,56789ABCdkmqrtuy, 9AJKPUVWflmot!''&,BDLQRSTUZbcgrs'(,CEMNOPQRfhipx$%',FGHIJKLMadegxz!(.

1=(0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0); 2=(0,0,1,−1,1,−1,0,0,0,0,0,0,0,0,0,0); 3=(0,0,0,1,0,−1,0,0,0,0,0,0,0,0,0,0); 4=(0,0,1,0,−1,0,0,0,0,0,0,0,0,0,0,0); 5=(0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0); 6=(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0); 7=(0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0); 8=(0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0); 9=(0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0); A=(0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0); B=(0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0); C=(0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0); D=(1,−1,1,0,1,0,0,0,0,0,0,0,0,0,0,0); E=(1,1,0,1,0,1,0,0,0,0,0,0,0,0,0,0); F=(1,1,0,−1,0,−1,0,0,0,0,0,0,0,0,0,0); G=(−1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0); H=(0,1,−1,1,0,0,1,0,0,0,0,0,0,0,0,0); I=(1,0,1,1,0,0,0,−1,0,0,0,0,0,0,0,0); J=(1,0,0,0,1,1,0,1,0,0,0,0,0,0,0,0); K=(0,1,0,0,−1,1,−1,0,0,0,0,0,0,0,0,0); L=(0,0,1,0,−1,0,1,1,0,0,0,0,0,0,0,0); M=(0,0,0,1,0,−1,−1,1,0,0,0,0,0,0,0,0); N=(1,0,1,0,0,−1,1,0,0,0,0,0,0,0,0,0); O=(0,−1,1,0,0,1,0,1,0,0,0,0,0,0,0,0); P=(−1,1,0,0,0,0,1,1,0,0,0,0,0,0,0,0); Q=(1,0,−1,−1,0,0,0,1,0,0,0,0,0,0,0,0); R=(0,1,1,−1,0,0,−1,0,0,0,0,0,0,0,0,0); S=(1,0,0,1,−1,0,−1,0,0,0,0,0,0,0,0,0); T=(0,1,0,1,1,0,0,1,0,0,0,0,0,0,0,0); U=(1,1,0,0,0,0,1,−1,0,0,0,0,0,0,0,0); V=(0,1,0,0,1,−1,−1,0,0,0,0,0,0,0,0,0); W=(1,0,0,0,−1,−1,0,1,0,0,0,0,0,0,0,0); X=(1,1,0,−1,0,1,0,0,0,0,0,0,0,0,0,0); Y=(1,−1,−1,0,1,0,0,0,0,0,0,0,0,0,0,0); Z=(0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0); a=(0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0); b=(0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0); c=(0,0,0,0,0,0,0,0,0,0,1,−1,0,0,0,0); d=(0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0); e=(0,0,0,0,0,0,0,0,1,0,−1,0,0,0,0,0); f=(0,0,0,0,0,0,0,0,1,−1,0,0,0,0,0,0); g=(0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0); h=(0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0); i=(0,0,0,0,0,0,0,0,1,1,−1,−1,0,0,0,0); j=(0,0,0,0,0,0,0,0,1,−1,1,−1,0,0,0,0); k=(0,0,0,0,0,0,0,0,1,−1,−1,−1,0,0,0,0); l=(0,0,0,0,0,0,0,0,1,1,1,−1,0,0,0,0); m=(0,0,0,0,0,0,0,0,1,1,−1,1,0,0,0,0); n=(0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0); o=(0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0); p=(0,0,0,0,0,0,0,0,0,0,0,1,−1,0,0,0,0); q=(0,0,0,0,0,0,0,0,0,1,0,−1,0,0,0,0); r=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0); s=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0); t=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0); u=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,0); v=(0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0); w=(0,0,0,0,0,0,0,0,0,0,0,0,1,0,−1,0); x=(0,0,0,0,0,0,0,0,0,0,0,0,1,−1,0,0); y=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1); z=(0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1); !=(0,0,0,0,0,0,0,0,0,0,0,0,1,1,−1,−1); ''=(0,0,0,0,0,0,0,0,0,0,0,0,1,−1,1,−1); #=(0,0,0,0,0,0,0,0,0,0,0,0,1,−1,−1,−1); $=(0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,−1); %=(0,0,0,0,0,0,0,0,0,0,0,0,1,1,−1,1); &=(0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1); '=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1); (=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1); )=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,−1)

### 14. 27D MMPH; 16 hyperedges

**141-16** 123456789ABCDEFGHIJKLMNOPQR,9STUVWXYZIabcdefghRijklmnop,Zqrstuvwxhyz!''#$%&p'()*-/:;, xvw<=>?21&$%@[\]BA;/:^`{KJ,91|S}4uVwIA~a+1D#d%RJ+2i+3M-l:,92+4+5=t6+6wIB+7+8[''F+9%RK +A+B_*O+C:, 9|3456+D+EYI~CDEF+F+GgR+2LMNO+H+Io,9+43456X+J+KI+7CDEFf+L+MR+ALMNOn+N+O, 9+5T+P5+6+Q+D+KI+8b+RE+9+S+F+MR+Bj+TN+C+U+H+O,9sT+P5+6W+J8I!b+RE+9e+LHR)j+TN+Cm+NQ, 9s=t6+6+Q+E7I![''F+9+S+GGR)_*O+C+U+IP,q+V+5+WT+P5+6wy+X+8+Yb+RE+9%'+Z+B+aj+TN+C:, +Vr+b|3456w+Xz+c~CDEF%+Z(+d+2LMNO:,+e+f+bSTUVZw+g+h+cabcdh%+i+j+dijklp:, +e+k<sU+ P>+lw+g+m@!c+R\+n%+i+o^)k+T`+p:,+f+k+4+W}3+l?w+h+m+7+Y+1C+n]%+j+o+A+a+3L+p{:.

1=(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), 2=(0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), 3=(0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), 4=(1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), 5=(1,−1,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), 6=(1,−1,−1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), 7=(1,−1,−1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), 8=(1,−1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), 9=(1,1,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), A=(1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), B=(0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), C=(0,0,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), D=(0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), E=(0,1,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), F=(1,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), G=(1,0,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), H=(1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), I=(0,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), J=(0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), K=(0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), L=(0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), M=(0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), N=(0,0,0,0,1,−1,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), O=(0,0,0,0,1,−1,−1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), P=(0,0,0,0,1,−1,−1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), Q=(0,0,0,0,1,−1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), R=(0,0,0,0,1,1,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), S=(0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), T=(0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), U=(0,0,0,0,0,0,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), V=(0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), W=(0,0,0,0,0,1,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), X=(0,0,0,0,1,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), Y=(0,0,0,0,1,0,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), Z=(0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), a=(0,0,0,0,0,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), b=(0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), c=(0,1,1,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), d=(0,1,−1,1,0,0,0,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), e=(0,1,1,−1,0,0,0,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), f=(0,1,1,1,0,0,0,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), g=(0,1,−1,−1,0,0,0,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), h=(0,1,1,−1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), i=(0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), j=(0,0,1,0,0,0,0,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), k=(0,0,0,1,0,0,0,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),

l=(0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
m=(0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
n=(0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
o=(0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
p=(0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
q=(0,0,0,0,0,0,0,0,0,1,−1,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
r=(0,0,0,0,0,0,0,0,0,1,−1,−1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
s=(0,0,0,0,0,0,0,0,0,1,−1,−1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
t=(0,0,0,0,0,0,0,0,0,1,−1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
u=(0,0,0,0,0,0,0,0,0,1,1,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
v=(0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
w=(0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
x=(0,0,0,0,0,0,0,0,0,0,0,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
y=(0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
z=(0,0,0,0,0,0,0,0,0,0,1,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
!=(0,0,0,0,0,0,0,0,0,1,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
''=(0,0,0,0,0,0,0,0,0,1,0,0,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
#=(0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
$=(0,0,0,0,0,0,0,0,0,0,1,−1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
%=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0),
&=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0),
'=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0),
(=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0),
)=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,1,−1,0,0,0,0,0,0,0,0,0,0),
*=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,−1,1,0,0,0,0,0,0,0,0,0,0),
−=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,−1,−1,0,0,0,0,0,0,0,0,0,0),
/=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,1,1,0,0,0,0,0,0,0,0,0,0),
:=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,−1,0,0,0,0,0,0,0,0,0,0),
;=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0),
<=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0),
==(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,0,0,0,0,0,0,0,0,0,0),
>=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0),
?=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,−1,0,0,0,0,0,0,0,0,0,0),
@=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,−1,0,0,0,0,0,0,0,0,0,0,0),
[=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,−1,0,0,0,0,0,0,0,0,0,0),
\=(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0),
]=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,0,0,0,0,0,0,0,0,0,0,0),
^=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0),
_=(0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0),
‘=(0,0,0,0,0,0,0,0,0,0,1,−1,1,0,0,0,0,−1,0,0,0,0,0,0,0,0,0),
{=(0,0,0,0,0,0,0,0,0,0,1,1,−1,0,0,0,0,−1,0,0,0,0,0,0,0,0,0),
|=(0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,−1,0,0,0,0,0,0,0,0,0),
}=(0,0,0,0,0,0,0,0,0,0,1,−1,−1,0,0,0,0,−1,0,0,0,0,0,0,0,0,0),
∼=(0,0,0,0,0,0,0,0,0,0,1,1,−1,0,0,0,0,1,0,0,0,0,0,0,0,0,0),
+1=(0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0),
+2=(0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,−1,0,0,0,0,0,0,0,0,0),
+3=(0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,−1,0,0,0,0,0,0,0,0,0),
+4=(0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0),
+5=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0),
+6=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0),
+7=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0),
+8=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0),
+9=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,1,−1,0,0,0,0,0),
+A=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,−1,1,0,0,0,0,0),
+B=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,−1,−1,0,0,0,0,0),
+C=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,1,1,0,0,0,0,0),
+D=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,−1,0,0,0,0,0),
+E=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0),
+F=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0),
+G=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,0,0,0,0,0),
+H=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0),
+I=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,−1,0,0,0,0,0),
+J=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,-1,0,0,0,0,0,0),
+K=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,-1,0,0,0,0,0),
+L=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0),
+M=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,0,0,0,0,0,0),
+N=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0),
+O=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0),
+P=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0),
+Q=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0),
+R=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,1,−1,0),
+S=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,−1,1,0),
+T=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,−1,−1,0),
+U=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,1,1,0),
+V=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,−1,0),
+W=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0),
+X=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0),
+Y=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,0),
+Z=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0),
+a=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,−1,0),
+b=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,−1,0,0),
+c=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,−1,0),
+d=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0),
+e=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,-1,0,0),
+f=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),
+g=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,1),
+h=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,1,0,0,0,0,−1),
+i=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,−1,0,0,0,0,−1),
+j=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,−1),
+k=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,−1,−1,0,0,0,0,-1),
+l=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,−1,0,0,0,0,1),
+m=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1),
+n=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,−1),
+o=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,−1),
+p=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1)

[1] M. Howard, J. Wallman, V. Veitech, and J. Emerson, Contextuality supplies the ‘magic’ for quantum computation, Nature (London) **510**, 351 (2014).

[2] S. D. Bartlett, Powered by magic, Nature (London) **510**, 345 (2014).

[3] A. Tavakoli and R. Uola, Measurement incompatibility and steering are necessary and sufficient for operational contextuality, Phys. Rev. Res. **2**, 013011 (2020).

[4] D. Saha, P. Horodecki, and M. Pawłowski, State independent contextuality advances one-way communication, New J. Phys. **21**, 093057 (2019).

[5] S. Kochen and E. P. Specker, The problem of hidden variables in quantum mechanics, J. Math. Mech. **17**, 59 (1967).

[6] E. Amselem, M. Rådmark, M. Bourennane, and A. Cabello, State-independent quantum contextuality with single photons, Phys. Rev. Lett. **103**, 160405 (2009).

[7] B. H. Liu, Y. F. Huang, Y. X. Gong, F. W. Sun, Y. S. Zhang, C. F. Li, and G. C. Guo, Experimental demonstration of quantum contextuality with nonentangled photons, Phys. Rev. A **80**, 044101 (2009).

[8] V. D'Ambrosio, I. Herbauts, E. Amselem, E. Nagali, M. Bourennane, F. Sciarrino, and A. Cabello, Experimental implementation of a Kochen-Specker set of quantum tests, Phys. Rev. X **3**, 011012 (2013).

[9] Y.-F. Huang, C.-F. Li, Y.-S. Zhang, J.-W. Pan, and G.-C. Guo, Experimental test of the Kochen-Specker theorem with single photons, Phys. Rev. Lett. **90**, 250401 (2003).

[10] G. Cañas, S. Etcheverry, E. S. Gómez, C. Saavedra, G. B. Xavier, G. Lima, and A. Cabello, Experimental implementation of an eight-dimensional Kochen-Specker set and observation of its connection with the Greenberger-Horne-Zeilinger theorem, Phys. Rev. A **90**, 012119 (2014).

[11] G. Cañas, M. Arias, S. Etcheverry, E. S. Gómez, A. Cabello, G. B. Xavier, and G. Lima, Applying the simplest Kochen-Specker set for quantum information processing, Phys. Rev. Lett. **113**, 090404 (2014).

[12] Y. Hasegawa, R. Loidl, G. Badurek, M. Baron, and H. Rauch, Quantum contextuality in a single-neutron optical experiment, Phys. Rev. Lett. **97**, 230401 (2006).

[13] H. Bartosik, J. Klepp, C. Schmitzer, S. Sponar, A. Cabello, H. Rauch, and Y. Hasegawa, Experimental test of quantum contextuality in neutron interferometry, Phys. Rev. Lett. **103**, 040403 (2009).

[14] G. Kirchmair, F. Zähringer, R. Gerritsma, M. Kleinmann, O. Gühne, A. Cabello, R. Blatt, and C. F. Roos, State-independent experimental test of quantum contextuality, Nature (London) **460**, 494 (2009).

[15] O. Moussa, C. A. Ryan, D. G. Cory, and R. Laflamme, Testing contextuality on quantum ensembles with one clean qubit, Phys. Rev. Lett. **104**, 160501 (2010).

[16] A. Cabello, Converting contextuality into nonlocality, Phys. Rev. Lett. **127**, 070401 (2021).

[17] R. Ramanathan, M. Rosicka, K. Horodecki, S. Pironio, M. Horodecki, and P. Horodecki, Gadget structures in proofs of the Kochen-Specker theorem, Quantum **4**, 308 (2020).

[18] K. Horodecki, M. Horodecki, P. Horodecki, R. Horodecki, M. Pawłowski, and M. Bourennane, Contextuality offers device-independent security, arXiv:1006.0468 [quant-ph].

[19] R. Ramanathan, Y. Liu, and P. Horodecki, Large violations in Kochen Specker contextuality and their applications, New J. Phys. **24**, 033035 (2022).

[20] A. Kulikov, M. Jerger, A. Potočnik, A. Wallraff, and A. Fedorov, Realization of a quantum random generator certified with the Kochen-Specker theorem, Phys. Rev. Lett. **119**, 240501 (2017).

[21] R. Gallego, N. Brunner, C. Hadley, and A. Acín, Device-independent tests of classical and quantum dimensions, Phys. Rev. Lett. **105**, 230501 (2010).

[22] S. Wehner, M. Christandl, and A. C. Doherty, Lower bound on the dimension of a quantum system given measured data, Phys. Rev. A **78**, 062112 (2008).

[23] A. Babazadeh, M. Erhard, F. Wang, M. Malik, R. Nouroozi, M. Krenn, and A. Zeilinger, High-dimensional single-photon quantum gates: Concepts and experiments, Phys. Rev. Lett. **119**, 180510 (2017).

[24] M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz, and A. Zeilinger, Automated search for new quantum experiments, Phys. Rev. Lett. **116**, 090405 (2016).

[25] M. Malik, M. Erhard, M. Huber, M. Krenn, R. Fickler, and A. Zeilinger, Multi-photon entanglement in high dimensions, Nat. Photon. **10**, 248 (2016).

[26] T. Zhong, H. Zhou, R. D. Horansky, C. Lee, V. B. Verma, A. E. Lita, A. Restelli, J. C. Bienfang, R. P. Mirin, T. Gerrits *et al.*, Photon-efficient quantum key distribution using time-energy entanglement with high-dimensional encoding, New J. Phys. **17**, 022002 (2015).

[27] C. Budroni, A. Cabello, O. Gühne, M. Kleinmann, and J.-Å. Larsson, Kochen-Specker contextuality, Rev. Mod. Phys. **94**, 045007 (2022).

[28] Z.-H. Liu, H.-X. Meng, Z.-P. Xu, J. Zhou, J.-L. Chen, J.-S. Xu, C.-F. Li, G.-C. Guo, and A. Cabello, Experimental test of high-dimensional quantum contextuality based on contextuality concentration, Phys. Rev. Lett. **130**, 240202 (2023).

[29] S. Paesani, J. F. F. Bulmer, A. E. Jones, R. Santagati, and A. Laing, Scheme for universal high-dimensional quantum computation with linear optics, Phys. Rev. Lett. **126**, 230504 (2021).

[30] D. Bacco, J. F. F. Bulmer, M. Erhard, M. Huber, and S. Paesani, Proposal for practical multidimensional quantum networks, Phys. Rev. A **104**, 052618 (2021).

[31] X. Zhan and L. Hu, Dimension-dependent noncontextuality inequalities with large contexts, Phys. Rev. A **104**, 032208 (2021).

[32] M. Frembs, S. Roberts, and S. D. Bartlett, Contextuality as a resource for measurement-based quantum computation beyond qubits, New J. Phys. **20**, 103011 (2018).

[33] M. Waegell and P. K. Aravind, Golay codes and quantum contextuality, Phys. Rev. A **106**, 062421 (2022).

[34] Y. Wang, Z. Hu, B. C. Sanders, and S. Kais, Qudits and high-dimensional quantum computing, Front. Phys. **8**, 589504 (2022).

[35] M. Planat and M. Saniga, Five-qubit contextuality, noise-like distribution of distances between maximal bases and finite geometry, Phys. Lett. A **376**, 3485 (2012).

[36] M. Waegell and P. K. Aravind, Parity proofs of the Kochen-Specker theorem based on the Lie algebra E8, J. Phys. A **48**, 225301 (2015).

[37] M. Pavičić, Arbitrarily exhaustive hypergraph generation of 4-, 6-, 8-, 16-, and 32-dimensional quantum contextual sets, Phys. Rev. A **95**, 062121 (2017).

[38] M. Waegell and P. K. Aravind, Minimal complexity of Kochen-Specker sets does not scale with dimension, Phys. Rev. A **95**, 050101(R) (2017), Supplemental Material.

[39] M. Pavičić and N. D. Megill, Vector generation of quantum contextual sets in even dimensional Hilbert spaces, Entropy **20**, 928 (2018).

[40] M. Pavičić, Hypergraph contextuality, Entropy **21**, 1107 (2019).

[41] M. Pavičić, M. Waegel, N. D. Megill, and P. Aravind, Automated generation of Kochen-Specker sets, Sci. Rep. **9**, 6765 (2019).

[42] M. Pavičić and N. D. Megill, Automated generation of arbitrarily many Kochen-Specker and other contextual sets in odd dimensional Hilbert spaces, Phys. Rev. A **106**, L060203 (2022).

[43] M. Pavičić, Quantum contextuality, Quantum **7**, 953 (2023).

[44] J. Zimba and R. Penrose, On Bell non-locality without probabilities: More curious geometry, Stud. Hist. Phil. Sci. **24**, 697 (1993).

[45] S. Matsuno, The construction of Kochen-Specker noncolourable sets in higher-dimensional space from corresponding sets in lower dimension: Modification of Cabello, Estebaranz and García-Alcaine′s method, J. Phys. A **40**, 9507 (2007).

[46] A. Cabello, J. M. Estebaranz, and G. García-Alcaine, Recursive proof of the Bell-Kochen-Specker theorem in any dimension $n > 3$, Phys. Lett. A **339**, 425 (2005).

[47] L. Lovász, On the Shannon capacity of a graph, IEEE Trans. Information Theory **25**, 1 (1979).

[48] M. Grötschel, L. Lovász, and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica **1**, 169 (1981).

[49] P. Lisoněk, P. Badziąg, J. R. Portillo, and A. Cabello, Kochen-Specker set with seven contexts, Phys. Rev. A **89**, 042101 (2014).

[50] http://puh.srce.hr/s/Qegixzz2BdjYwFL.

[51] M. Pavičić, N. D. Megill, and J.-P. Merlet, New Kochen-Specker sets in four dimensions, Phys. Lett. A **374**, 2122 (2010).

[52] A. Peres, Two simple proofs of the Bell-Kochen-Specker theorem, J. Phys. A **24**, L175 (1991).

[53] A. Cabello, J. M. Estebaranz, and G. García-Alcaine, Bell-Kochen-Specker theorem: A proof with 18 vectors, Phys. Lett. A **212**, 183 (1996).

[54] M. Waegell and P. K. Aravind, Parity proofs of the Kochen-Specker theorem based on 24 rays of Peres, Found. Phys. **41**, 1785 (2011).

[55] M. Pavičić, J.-P. Merlet, B. D. McKay, and N. D. Megill, Kochen-Specker vectors, J. Phys. A **38**, 1577 (2005).

[56] M. Pavičić and N. D. Megill, Vector generation of contextual sets, EPJ Web. Conf. **198**, 00009 (2019).

[57] P. Lisoněk, Kochen-Specker sets and Hadamard matrices, Theor. Comp. Sci. **800**, 142 (2019).

[58] H. Bechmann-Pasquinucci and W. Tittel, Quantum cryptography using larger alphabets, Phys. Rev. A **61**, 062308 (2000).

[59] A. Cabello, V. D’Ambrosio, E. Nagali, and F. Sciarrino, Hybrid ququart-encoded quantum cryptography protected by Kochen-Specker contextuality, Phys. Rev. A **84**, 030302(R) (2011).

[60] K. Svozil, Bertlmann′s chocolate balls and quantum type cryptography, in *Physics and Computation 2010*, edited by H. Guerra (Department of Mathematics, University of Azores, Ponta Delgada, Portugal, 2010), pp. 235–249.

[61] M. Krenn, M. Huber, R. Fickler, R. Lapkiewicz, S. Ramelow, and A. Zeilinger, Generation and confirmation of a $(100\times100)$-dimensional entangled quantum system, Proc. Natl. Acad. Sci. USA **111**, 6243 (2014).