INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **38** (2005) 1577–1592 doi:10.1088/0305-4470/38/7/013

Kochen–Specker vectors

Mladen Pavičić[1], **Jean-Pierre Merlet**[2], **Brendan McKay**[3] **and Norman D Megill**[4]

[1] University of Zagreb, Gradjevinski fakultet, PO Box 217, HR-10001 Zagreb, Croatia
[2] INRIA, projet COPRIN: 06902 Sophia Antipolis Cedex, France
[3] Department of Computer Science, Australian National University, Canberra, ACT, 0200, Australia
[4] Boston Information Group, 30 Church St, Belmont, MA 02478, USA

E-mail: pavicic@grad.hr, Jean-Pierre.Merlet@sophia.inria.fr, bdm@cs.anu.edu.au and nm@alum.mit.edu

Received 3 September 2004, in final form 22 November 2004
Published 2 February 2005
Online at stacks.iop.org/JPhysA/38/1577

Abstract

We give a constructive and exhaustive definition of Kochen–Specker (KS) vectors in a Hilbert space of any dimension as well as of all the remaining vectors of the space. KS vectors are elements of any set of orthonormal states, i.e., vectors in an n-dimensional Hilbert space, $\mathcal{H}^n, n \geqslant 3$, to which it is impossible to assign 1s and 0s in such a way that no two mutually orthogonal vectors from the set are both assigned 1 and that not all mutually orthogonal vectors are assigned 0. Our constructive definition of such KS vectors is based on algorithms that generate MMP diagrams corresponding to blocks of orthogonal vectors in $\mathbb{R}^n$, on algorithms that single out those diagrams on which algebraic 0–1 states cannot be defined, and on algorithms that solve nonlinear equations describing the orthogonalities of the vectors by means of statistically polynomially complex interval analysis and self-teaching programs. The algorithms are limited neither by the number of dimensions nor by the number of vectors. To demonstrate the power of the algorithms, all four-dimensional KS vector systems containing up to 24 vectors were generated and described, all three-dimensional vector systems containing up to 30 vectors were scanned, and several general properties of KS vectors were found.

PACS numbers: 03.65.Ta, 03.65.Ud

1. Introduction

Recently proposed experimental tests of the Kochen–Specker (KS) theorem [1, 2], skepticism on the feasibility of such experiments [3–7], positive experiments recently carried out [8] and recent theoretical elaborations on the theorem [9–22] prompted a renewed interest in the KS theorem.

The KS theorem proves that there is a set of measurements that can be carried out on a finite-dimensional quantum system in such a way that if one assumed that the values of measured observables are completely independent of all other observables that can be measured on the same system, then one would run into a contradiction. Hence, a quantum system cannot possess a definite value of a measurable property prior to measurement, and quantum measurements (essentially detector clicks) carried out on quantum systems cannot be ascribed predetermined values (say 0 and 1). To arrive at the claim, one considers an orthonormal set of states $\{\psi_1, \ldots, \psi_n\}$, i.e., vectors in an n-dimensional Hilbert space, $\mathcal{H}^n, n \geqslant 3$. Projectors onto these states satisfy $\sum_{i=1}^{n} P_i = I$, where $P_i = \psi_i \psi_i^\dagger$. Now, Kochen and Specker proved [23] that there is no function $f : \mathcal{H} \to \mathbb{R}$ satisfying the sum rule $\sum_{i=1}^{n} f(P_i) = f\big(\sum_{i=1}^{n} P_i\big) = f(I)$ for all sets of projectors P_i. Hence, there is at least one set of projectors $\{P_i, P_i', \ldots\}$ and the corresponding set of vectors $\{\psi_i, \psi_i', \ldots\}$ for which the sum rule is not satisfied. Choosing $f(P_i) \in \{0, 1\}$ $(f(I) = 1)$, the theorem amounts to the following claim: in $\mathcal{H}^n, n \geqslant 3$, it is impossible to assign 1s and 0s to all vectors from such a set—which we call a *KS set*—in such a way that [24]

(i) no two orthogonal vectors are both assigned the value 1;
(ii) in any subset of n mutually orthogonal vectors, not all of the vectors are assigned the value 0.

All the vectors from a KS set, as defined above, we call *KS vectors*. KS vectors in each KS set form subsets of n mutually orthogonal vectors. We arrive at one subset from another by a series of rotations in two-dimensional planes around $(n-2)$-dimensional subspaces as explained in section 4. Thus, any two subsets share at least one vector which is orthogonal to all other vectors in both subsets and in an n-dimensional space, two subsets can share up to $n-2$ vectors. The KS vectors correspond to the directions of the quantization axes of the measured eigenstates within experiments which have no classical counterparts, and when we speak of finding KS vectors we mean finding these directions. We stress here that it is not our aim to give yet another proof of the KS theorem but to determine the class of all KS vectors from an arbitrary $\mathcal{H}^n$ as well as the class of all *non-KS vectors*, i.e., vectors from the remaining sets of vectors from $\mathcal{H}^n$. By the class of non-KS vectors we mean vectors that allow 0–1 states and that correspond to the directions of the quantization axes of the measured eigenstates within experiments which do have classical counterparts and when we speak of finding non-KS vectors we mean finding the latter directions.

The original KS theorem [23] made use of 192 (claimed 117) three-dimensional vectors. Subsequent attempts to reduce the number of vectors gave the following minimal results (usually called *records*): Bub's system contains 49 vectors (claimed 33) [25], Conway–Kochen's has 51 (claimed 31) [26, p 114], and Peres' system has 57 (claimed 33) [27] three-dimensional vectors[5]; Kernaghan's system contains 20 four-dimensional vectors with the smallest loops (see the definition below) of size 2 [29]; Cabello's system has 18 four-dimensional vectors with the smallest loops of size 3 [30], etc. Reducing the number of vectors is important for devising experimental setups [16], especially so as recently a single qubit KS scheme was formulated [9] by means of auxiliary quantum systems (ancillas) of the measuring apparatus and subsequently connected with the original KS formulation [10]. On the other hand, knowing the class of all KS vectors is important for better theoretical insight

[5] The reasons why Kochen–Specker's, Bub's, Conway–Kochen's and Peres' systems should be considered as 192, 49, 51 and 57 and not as 117, 33, 31 and 33 vector systems, respectively, are given in section 5-(xi) in accordance with the results independently obtained by Larsson [28].

into the quantum theory and possibly designing quantum computers. However, no general method for constructing sets of KS vectors has been proposed so far and the aim of this paper is to give one. In doing so we will follow the ideas put forward in [31–33].

So far, KS vectors have been constructed either by means of partial Boolean algebras and orthomodular lattices [23, 34, 38, 39], by direct experimental proposals [1, 2, 16], or by combining rays in $\mathbb{R}^n$ [25, 27, 29]. These approaches have two disadvantages: first, they depend on human ingenuity to find ever new examples and 'records', and second, their complexity grows exponentially with increasing numbers of dimensions and vectors. For example, lattices of orthogonal n-tuples have 2^n elements (Hasse diagrams) [40] and, on the other hand, the complexity of nonlinear equations describing combinations of orthogonalities also grows exponentially.

As opposed to this, we are able to give algorithms for generation of all the equations that have KS vectors as their solutions and to effectively solve them (up to a reasonably chosen number of vectors and dimensions—limited only by the speed of today's computers) in a way that is essentially of a statistically polynomial complexity. We first recognize that a description of a discrete observable measurement (e.g., spin) in $\mathcal{H}^n$ can be rendered as a 0–1 measurement of the corresponding projector along the vector in $\mathbb{R}^n$ onto which the projector projects. Hence, we deal with orthogonal triples in $\mathbb{R}^3$, quadruples in $\mathbb{R}^4$, etc, which correspond to possible experimental designs, and to find KS vectors means finding such n-tuples in $\mathbb{R}^n$.

The orthogonalities of vectors within these n-tuples can be described by nonlinear equations of the type given in equation (1) that have solutions. There are however billions of such nonlinear equations that have no solutions even for the smallest KS sets. And their number grows exponentially with the increase of both the number of KS vectors and the dimension of their space. So, we established a one-to-one correspondence between nonlinear equations and graphs (MMP diagrams). We can handle graphs exponentially faster than nonlinear equations but there are nevertheless billions of them. Therefore, we designed a self-teaching generation algorithm for MMP diagrams: graphs containing subgraphs that correspond to equations that cannot have a solution are not generated. This reduces the generation complexity to a statistically polynomial one and the time required for obtaining the MMP diagrams corresponding to systems of nonlinear equations with solutions from billions of years to hours and days.

To switch back from the MMP diagrams to nonlinear equations to solve them at this stage would again take quite some time. Therefore we defined the notion of an *algebraic dispersion-free state* (0–1 state) on MMP diagrams. It turns out that only a small percentage of the obtained MMP diagrams cannot have 0–1 states. Their direct verification is again of exponential complexity. So, we developed algorithms with backtracking that discard MMP diagrams with 0–1 states and whose complexity turns out to be statistically polynomial.

The diagrams finally obtained correspond to candidate sets of nonlinear equations that contain KS sets provided the equations have real solutions. Algorithms for solving nonlinear equations, such as the Gröbner basis and homotopy, are also mostly of at least exponential complexity and have been tested without success on representative systems. Therefore we designed new ones based on interval analysis and Ritt's characteristic set calculations and were able to reduce their complexity to a statistically polynomial one. This rounds up the constructive and exhaustive definition of KS sets and vectors and makes their generation feasible for reasonably chosen numbers of vectors and dimensions.

The paper is organized as follows. In section 2, MMP diagrams are defined and the algorithms as well as the programs for their generation are presented. In section 3, we give the algorithm and program for finding whether MMP diagrams can be assigned a set of dispersion-free 0–1 states and determining the latter sets when there is at least one set of 0–1 states together with the smallest MMP diagrams that do not allow `0–1` states. In section 4, we establish a link between MMP diagrams that do not allow `0–1` states and systems of nonlinear equations whose solutions are the KS vectors. We then give the algorithms and methods for solving the equations in a statistically polynomial time. In section 5, we present the new results we obtained.

2. MMP diagrams

We start by describing vectors as vertices (points) and orthogonalities between them as edges (lines connecting vertices), thus obtaining MMP diagrams [31, 33, 35] which are defined as follows:

1. every vertex belongs to at least one edge;
2. every edge contains at least three vertices;
3. edges that intersect each other in $n-2$ vertices contain at least n vertices.

Isomorphism-free generation of MMP diagrams follows the general principles established by [36], which we now recount briefly.

Deleting an edge from an MMP diagram, together with any vertices that lie only on that edge, yields another MPP diagram (perhaps the vacuous one with no vertices). Consequently, every MMP diagram can be constructed by starting with the vacuous diagram and adding one edge at a time, at each stage having an MMP diagram.

We can represent this process as a rooted tree whose vertices correspond to MMP diagrams whose vertices and edges have unique labels. The vacuous diagram is at the root of the tree, and for any other diagram its parent node is the diagram formed by deleting the edge with the highest label. The isomorph rejection problem is to prune this tree until it contains just one representative of each isomorphism class of diagram. This can be achieved by the application of two *rules*.

Given a diagram D, we can identify the valid positions to add a new edge such that conditions 2–3 are enforced. According to the symmetries of D, some of these positions are equivalent. The first rule is that exactly one position in each equivalence class of positions is used; a node in the tree formed by adding an edge in any other position is deleted together with all its descendants.

To understand the second rule, consider a diagram D' with at least one edge. We label the edges of D' in a canonical order, which is an order independent of any previous labelling. Then we define the *major class* of edges as those that are equivalent under the symmetries of D' to the edge that is last in canonical order. The second rule is that when D' is constructed by adding an edge e to a smaller diagram, delete D' (and all its descendants) unless e is in the major class of edges of D'.

According to the theory in [36], application of both rules together is sufficient: exactly one diagram from each isomorphism class remains in the tree. Our implementation used `nauty` [37] for computing symmetries and canonical orderings. The method allows for very efficient parallelization of the computation. A generation tree for MMP diagrams with nine vertices and the smallest loop of size 5 is shown in figure 1.

MMP diagrams with three vertices per edge and with smallest loops (edge polygons) of size 5 graphically resemble Greechie diagrams [38]. Greechie diagrams are a handy way to draw Hasse diagrams that represent orthomodular lattices. The complexity of Hasse diagrams grows exponentially with increasing dimensions and the smallest loops of the corresponding are of size 5, while MMP diagrams allow loops of sizes 4, 3 and 2 (two edges share at least

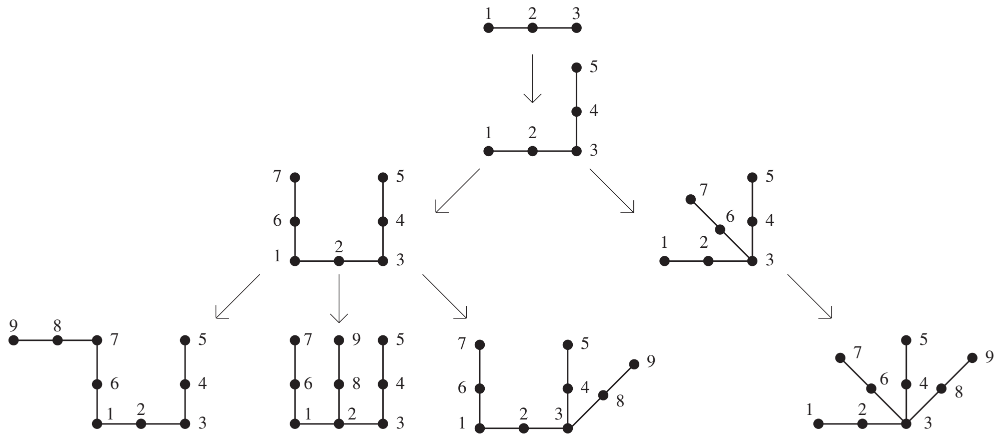

Figure 1. An example of a generation tree for connected MMP diagrams: nine vertices and the smallest loop of size 5 (for nine vertices a loop cannot be formed; the first loop appears with ten vertices: 123, 345, 567, 789, 9A1). (cf [33, 35]).

two vertices). Besides, it would be quite a challenge to find a direct lattice representation of KS vectors.

We denote vertices of MMP diagrams by 1, 2, . . . , A, B, . . . , a, b, ... By the above algorithm we generate MMP diagrams with chosen numbers of vertices and edges and a chosen minimal loop size. For example, in the examples given in figure 2 we generate diagrams with four vertices within an edge and minimal loops of sizes 2 and 3. Our programs handle diagrams with up to 90 vertices, but this limit could easily be extended.

3. Algebraic states on MMP diagrams

To find diagrams that cannot be ascribed 0–1 values we apply an algorithm which we call states01. The algorithm is an exhaustive search of MMP diagrams with backtracking. The criterion for assigning 0–1 (dispersion-free) states is that each edge must contain exactly one vertex assigned to 1, with the others assigned to 0. As soon as a vertex on an edge is assigned a 1, all other vertices on that edge become constrained to 0, and so on. The algorithm scans the vertices in some order, trying 0 then 1, skipping vertices constrained by an earlier assignment. When no assignment becomes possible, the algorithm backtracks until all possible assignments are exhausted (no solution) or a valid assignment is found. In principle the algorithm is exponential, but because the diagrams of interest are tightly coupled, constraints build up quickly. For the range of diagram size in our study, we found that the average time per diagram appeared to grow polynomially with the diagram size.

To implement the algorithm we wrote a program that selects MMP diagrams with three and four vertices per edge on which 0–1 states cannot be defined. The smallest such diagrams are given in figure 2.

- *three vertices per edge*
  seven vertices—five edges (smallest loops of size 3): 123,345,561,275,476 (triangle);
  15-11 (4): 123,345,567,789,9AB,BC1,CD6,2DA,2E8,4FA,CEF (hexagon),
  123,345,567,789,9AB,BCD,DE1,4AE,28C,2FA,6FD (heptagon);
  19-13 (5): 123,345,567,789,9AB,BCD,DEF,FG1,2IA,6IE,4HC,8JG,HIJ figure 2(*d*),
  123,345,567,789,9AB,BCD,DE1,EI7,2F9,4GB,IJG,FJH,CH6 (heptagon);

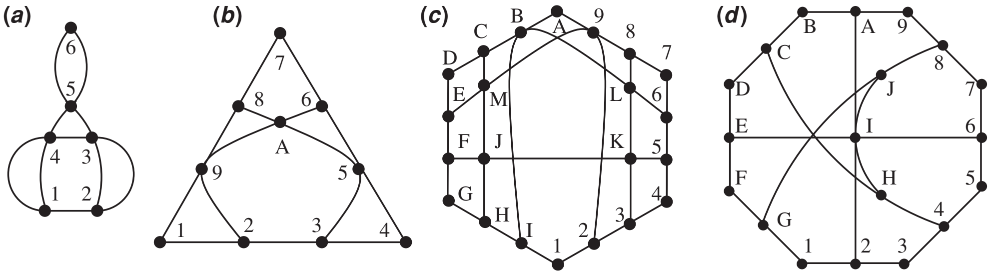

Figure 2. Smallest MMP diagrams without 0–1 states: (1) four vertices per edge: (*a*) loops of size 2: six vertices—three edges; (*b*) loops of size 3: 10-5; (*c*) loops of size 4: 22-11; (2) three vertices per edge: (*d*) loops of size 5: one of two 19-13; the other is shown in figure 2(*b*) of [33].

- *four vertices per edge*
  6-3 (smallest loops of size 2): `1234,2356,1456` figure 2(*a*);
  10-5 (smallest loops of size 3): `1234,4567,7891,35A8,29A6` figure 2(*b*);
  22-11 (smallest loops of size 4):
  `1234,4567,789A,ABCD,DEFG,GHI1,FJK5,HJMC,3KL8,IBL6,29ME.` Figure 2(*c*),
  `1234,4567,789A,ABCD,DEF1,FGH5,EMJ6,2GLC,3IJ8,HIKB,MLK9` (pentagon);
  38-19 (5): `1234,1567,289A,5BCD,8BEF,3GHI,6JKL,GJMN,CHOP,EMQR,OQST,RUVW,`
  `4UXY,9SZa,FIbc,KTXb,7VZc,ALPW,DNYa` (dodecagon).

Further details of the algorithm and the program `states01` will be given elsewhere [41].

4. Kochen–Specker vectors

To find KS vectors we follow the idea put forward in [31, 33] and proceed so as to require that their number, i.e. the number of vertices within edges, corresponds to the dimension of $\mathbb{R}^n$ and edges correspond to $n(n-1)/2$ equations resulting from inner products of vectors being equal to zero which means orthogonality. So, e.g., an edge of length 4, `BCDE`, represents the following six equations:

$$
\begin{aligned}
\mathbf{a}_B \cdot \mathbf{a}_C &= a_{B1}a_{C1} + a_{B2}a_{C2} + a_{B3}a_{C3} + a_{B4}a_{C4} = 0,\\
\mathbf{a}_B \cdot \mathbf{a}_D &= a_{B1}a_{D1} + a_{B2}a_{D2} + a_{B3}a_{D3} + a_{B4}a_{D4} = 0,\\
\mathbf{a}_B \cdot \mathbf{a}_E &= a_{B1}a_{E1} + a_{B2}a_{E2} + a_{B3}a_{E3} + a_{B4}a_{E4} = 0,\\
\mathbf{a}_C \cdot \mathbf{a}_D &= a_{C1}a_{D1} + a_{C2}a_{D2} + a_{C3}a_{D3} + a_{C4}a_{D4} = 0,\\
\mathbf{a}_C \cdot \mathbf{a}_E &= a_{C1}a_{E1} + a_{C2}a_{E2} + a_{C3}a_{E3} + a_{C4}a_{E4} = 0,\\
\mathbf{a}_D \cdot \mathbf{a}_E &= a_{D1}a_{E1} + a_{D2}a_{E2} + a_{D3}a_{E3} + a_{D4}a_{E4} = 0.
\end{aligned}
\qquad (1)
$$

Each possible combination of edges for a chosen number of vertices corresponds to a system of such nonlinear equations. A solution to systems which correspond to MMP diagrams without 0–1 states is a set of components of KS vectors we want to find. Thus the main clue to finding *all* KS vectors is the exhaustive generation of all MMP diagrams as given in section 2, then picking out all those diagrams that cannot have 0–1 states as presented in section 3, establishing the correspondence between the latter diagrams and the equations for the vectors as shown in equation (1), and finally solving the systems of the so obtained equations.

In practice, we actually merge these four stages so as to avoid generating those diagrams that cannot have a solution[6]. For systems of equations of the type given by equation (1) that do

[6] This merging is crucial. Without it we would not be able to reduce the exponential complexity of the problem to the statistically polynomial one.

have solutions that do not allow 0–1 states, such solutions are KS vectors that correspond to vertices of MMP diagrams. Mutually orthogonal vectors correspond to edges, and connected edges, i.e., MMP diagrams themselves correspond to the systems of equations. For instance, in the connected edges 1234,4567 vectors 1,2,3,4 and 4,5,6,7, are mutually orthogonal and 4567 is obtained from 1234 by four-dimensional rotations (1234 and 4567 are connected by 4). A general two-dimensional rotation is a rotation by an angle around a fixed point in the same plane. A general three-dimensional rotation is a rotation in a two-dimensional plane by an angle around a fixed axis perpendicular to this plane. So, we define a general four-dimensional rotation as a rotation in a two-dimensional plane by an angle around a fixed two-dimensional plane. What is common to all these rotations is that they always take place in a two-dimensional plane. Hence, we define an n-dimensional rotation as a rotation in a two-dimensional plane by an angle around a fixed $(n-2)$-dimensional subspace [41]. This also explains the case of a smallest loop of size 2 in the four-dimensional case. For example, we arrive at 4561 from 1234 by a rotation in the two-dimensional plane determined by the vectors 2,3 (and also by the vectors 5,6) around the plane determined by the vectors 1,4.

Finding KS vectors is not a well-posed problem in terms of solving, though. Indeed if $\mathcal{V}$ is a KS vector then $\lambda\mathcal{V}$ is also a KS vector for any non-zero scalar λ. Furthermore, if $\mathcal{S}$ is a set of KS vectors, then $\mathcal{RS}$ is also such a set for any arbitrary rotation matrix $\mathcal{R}$. We may simplify the problem by considering only unit vectors (i.e. vectors whose Euclidean norm is 1 and hence vectors whose components have a value in the range $[-1, 1]$). To avoid the rotation problem, we may assume that one n-tuple is the orthonormal basis of $\mathbb{R}^n$. Under these assumptions, some of the orthogonality equations simplify. For example, if 1234 is the basis of $\mathbb{R}^4$ with 1 = [0, 0, 0, 1] then 1567 indicates that the fourth components of 5,6,7 are 0. The non-collinearity constraint also plays an important role. For example, 1235 is not a possible n-tuple as three components of 5 would be 0 and hence 5 would be collinear with 4.

This has prompted us to develop a *preliminary pass*, which allows elimination of n-tuples that cannot lead to a solution. Consider a system of m four-dimensional vectors. The preliminary pass makes use of an $m \times 4$ table $\mathcal{T}$, called the 0-*table*, with an entry set to 1 when a vector component cannot be 0. For example, if vector j has components $[a_{j1}, 0, 0, a_{j4}]$, then neither a_{j1} nor a_{j4} can be 0 (otherwise the vector will be collinear with one of the vectors of the basis) and $\mathcal{T}[j, 1] = \mathcal{T}[j, 4] = 1$. The preliminary pass selects a set of four four-dimensional vectors as the basis of $\mathbb{R}^4$. It then applies a set of simplification rules on the the orthogonality equations. For example, if the equation is $a_{jk}a_{ik} = 0$ and $\mathcal{T}[j, k] = 1$ then a_{ik} is set to 0. Each time a vector component value is determined the 0-table is updated and the preliminary pass is restarted. The process will stop when no further simplification may be performed or when a constraint violation occurs (e.g., one equation implies that a_{jk} should be 0 while the 0-table indicates that this component cannot be 0) in which case the system cannot have a solution. The simplification rules used in the preliminary pass depend on the space dimension.

The preliminary pass has been implemented as a C program that has been added as a filter in the generation program. For avoiding the exponential growth of the number of generated MMP diagrams it is essential that the candidate KS-sets should be generated incrementally, i.e. that the program generates sequentially all systems starting with a given m n-tuples before modifying the mth n-tuple. By using this incremental generation during the preliminary pass determines that an initial set of m n-tuples has no solution and that no further systems starting with this set will be generated. For example, for 18 vectors and 12 quadruples, without such a filter we would generate $>2.9 \times 10^{16}$ systems—what would require more than 30 million years on a 2 GHz CPU—while the filter reduces the generation to 100 220 systems (obtainable within <30 min on a 2 GHz CPU). Thereafter `states01` gives us 26 800 systems without 0–1 states in <5 s.

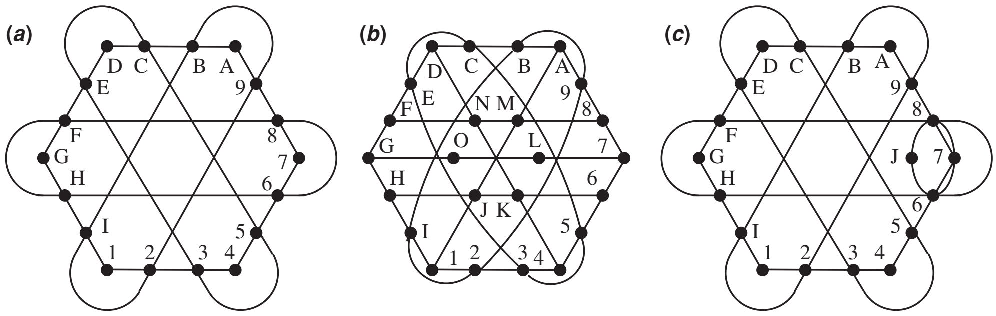

Figure 3. Smallest four-dimensional KS systems with (1) loops of size 3: (*a*) 18-9 (isomorphic to Cabello *et al* [30]); (*b*) 24(22)-13 not containing system (*a*), with values $\notin \{-1, 0, 1\}$; (2) loops of size 2: (*c*) 19(18)-10.

For the remaining systems, two solvers have been developed. One is based on a specific implementation of the Ritt characteristic set calculation [43]. Assume that a vector $\mathtt{V} = [a_{V1}, 0, 0, a_{V4}]$ (this implies that a_{V1}, a_{V4} cannot both be 0) is orthogonal to $\mathtt{W} = [a_{W1}, 0, a_{W2}, a_{W4}]$. From the orthogonality condition $a_{W1}a_{V1} + a_{W4}a_{V4} = 0$, we deduce that $a_{W1} = -a_{W4}a_{V4}/a_{V1}$ (as a_{V1} cannot be 0) and that a_{W4} cannot be 0. This information is propagated to the other equations and, as for the preliminary pass, simplification rules are applied to the equations, allowing us to determine further unknowns and to update the 0-table. The process is repeated until no further unknowns can be determined or until a constraint violation occurs. If no violation occurs, we will usually get a set of remaining equations that is quite simple and that allows us to determine all solutions. This solver has been implemented using the symbolic computation software Maple. For example, for system (*a*) in figure 3: `1234,4567,789A,ABCD,DEFG,GHI1,35CE,29BI,68FH`, we get (in <10 s on a 2 GHz CPU) the remaining set of ten equations: $2a_{62}^2 = 2a_{C1}^2 = 2a_{G3}^2 = 2a_{54}^2 = 4a_{E4}^2 = 1, 2a_{53}^2 = 2a_{I2}^2 = 2a_{94}^2 = 4a_{A2}^2 = 2a_{I1}^2 = 1$ from the roots of which we may deduce the other vector components (e.g., we get 6 as $a_{62}[0, 1, -2a_{54}a_{94}a_{53}, a_{94}]$). The drawback of this approach is that it does not always allow us to completely solve the equational systems: we may end up with a system with fewer equations for which no further constraints can be propagated.

Our second solver is based on interval analysis. An interval evaluation of an equation $f(x_1, \ldots, x_m) = 0$ is a range $F = [a, b]$ such that if all the unknowns $x_1, \ldots, x_m$ are restricted to lie within given ranges, then whatever the values of the unknowns in their range we have $a \leqslant f(x_1, \ldots, x_n) \leqslant b$. A simple way to calculate an interval evaluation is to use interval arithmetic that simply replaces all mathematical operators by an interval equivalent. For example, the interval evaluation of the orthogonality condition $x_1y_1 + x_2y_2$ with $x_1, x_2 \in [0.5, 1]$ and $y_1 \in [0.1, 0.2], y_2 \in [0.2, 1]$ is calculated as $[0.5, 1][0.1, 0.2] + [0.5, 1][0.2, 1] = [0.05, 0.2] + [0.01, 1] = [0.06, 1.2]$. Note that if the interval evaluation of an equation does not include 0 then there is no value of the unknowns in their range that can cancel the equation. A *box* will be a set of ranges, one for each unknown.

Solving a KS system is an appropriate problem for interval analysis, since all the unknowns are in the range $[-1, 1]$. The set of these unknowns is the box $\mathcal{B}_0$. The system of equations to be solved consists of the equations derived from the orthogonality conditions between the vectors and of the *unitary equations* that describe that each vector is a unit vector.

A basic solver uses a list of boxes $\mathcal{L}$ that initially have element $\mathcal{B}_0$. At step i, the algorithm processes box $\mathcal{B}_i$ of $\mathcal{L}$ and calculates the interval evaluation of the orthogonality and unitary equations: if the interval evaluation of one of these equations does not include 0, then the algorithm will process the next box in the list. Otherwise two new boxes will be generated from $\mathcal{B}_i$ by bisecting the range of the box. These boxes will be added to the list, and the next box in the list will be processed. The algorithm will stop either when all the boxes have been processed (meaning the system has no solution) or when the width of all the ranges in a box is less than a small value while the interval evaluations of all the equations still include 0 (meaning a solution is obtained). Note that the method is mostly sensitive to the number of unknowns (which explains why the vectors that appear only once in the KS system should be eliminated) and not so much to the number of equations. In contrast, additional equations may even reduce the computation time. For example, consider a triplet $\mathbf{X_i X_j X_k}$ in 3D: using the orthogonality condition, $\mathbf{X_k}$ is obtained as $\mathbf{X_k} = \pm\mathbf{X_i} \times \mathbf{X_j}$. We get therefore two possible solutions for $\mathbf{X_k}$, and additional equations will be obtained by writing that the square of each component of $\mathbf{X_k}$ should be equal to the square of the same component of $\mathbf{X_i} \times \mathbf{X_j}$.

Numerous methods may be used to improve the efficiency of the basic solver (especially to prove that indeed a system has a solution). We use the interval analysis library `ALIAS`[7] to deal efficiently with the KS systems.

Interval analysis has in principle an exponential complexity, due to the bisection process. But it has been experimentally shown that in some cases, the practical complexity is only polynomial. According to our tests (over 400 billion systems have been checked) it appears that the solving of the KS systems has indeed only a statistically polynomial complexity. It must also be noted that the solver may be used during the generation of the MMP diagrams as a complement to the preliminary pass to avoid the exponential growth of the number of generated MMP diagrams. Indeed, for a diagram that has not been rejected by the preliminary pass, we may run the solver to further check if the diagram has a solution. But since the solver may be relatively computer intensive, we have to use an adaptive version in which the number of allowed bisections is limited. For example this number may be large for relatively small sub-graphs because determining that they do not have a solution allows us to avoid the generation of a large number of diagrams. On the other hand, the number of allowed bisections will be small for sub-graphs whose size is close to the maximum (and consequently from which few diagrams will be deduced), thus avoiding increased generation time.

We also developed a checking program that finds solutions from assumed sets, say $\{-1, 0, 1\}$, even faster ($<$1 s on a 2 GHz CPU) by precomputing all possible scalar products. The main algorithm scans the vertices and tries to assign unique vectors to them so that all vectors assigned to a given edge are orthogonal. In the case of a conflict the algorithm backtracks, until either all possible assignments have been exhausted or a solution is found. We match its exponential behaviour by scanning next those vertices most tightly coupled to those already scanned, helping to force conflicts to show up early on so that backtracking can take care of them more quickly.

Further details of the algorithms and programs presented in this section will be given elsewhere [41].

5. New results and conclusions

In this paper we presented algorithms that generate and those that solve sets of arbitrarily many Kochen–Specker (KS) vectors that are of polynomial complexity or at least of statistically polynomial complexity. The algorithms merge generation of MMP diagrams corresponding to blocks of orthogonal vectors in $\mathbb{R}^n$ (section 2), singling out MMP diagrams on which `0-1` states cannot be defined (section 3), and solving nonlinear equations describing the orthogonalities of the vectors by means of interval analysis (section 4), so as to eventually

[7] www.inria.fr/coprin/logiciels/ALIAS/ALIAS.html.

generate KS vectors in a statistically polynomially complex way. Using the algorithms we obtained the following results:

(i) A general feature we found to hold for all MMP diagrams without 0–1 states we tested is that the number of edges, b, and the number of vertices that share more than one edge, a^*, satisfy the following inequality: $nb \geqslant 2a^*$, where n is the number of vertices per edge. Hence, there are no KS vectors that share at least two of b n-tuples in their KS set whose number $a^* > \frac{nb}{2}$. In $\mathbb{R}^n$ this means that we cannot arrive at systems with more unknowns than equations when we disregard the unknowns that appear in only two equations. To prove the feature for an arbitrary n remains an open problem.

(ii) For MMP diagrams without 0–1 states with three vertices per edge and $a < 30$ as well as with four vertices per edge and $a < 23$ the stronger inequality holds: $nb \geqslant 2a$. The only exception to this rule we have found is the original Kochen–Specker system with 192 vertices (see (xi) and figure 6). At what a for a chosen n this inequality ceases to hold is an open problem.

(iii) None of the systems corresponding to the smallest diagrams without 0–1 states given in section 3 and figure 2 has a solution[8]. The smallest KS vectors that we found to have real solutions are presented in figures 3 and 4.

(iv) Between the four-dimensional system shown in figure 3(*a*) and the one shown in figure 3(*b*) (both with smallest loops of size 3) there are 62 systems with loops of size 3, all containing the system (*a*), 37 of which do not have solutions from $\{-1, 0, 1\}$. System (*b*) is the first system with loops of size 3 not containing (a): `1234`, `4567`, `789A`, `ABCD`, `DEFG`, `GHI1`, `FNM8`, `GOL7`, `HJK6`, `DNK4`, `AMJ1`, `35CE`, `B29I`. One of its solutions is `12...NO` = {1,0,1,1} {1,0,−2,1} {1,0,0,−1} {0,1,0,0} {0,0,1,0} {0,0,0,1} {1,0,0,0} {0,2,2,1} {0,2,−1,−2} {0,1,−2,2} {3,2,2,1} {1,−2,0,1} {−1,0,1,1} {1,1,0,1} {1,−1,1,0} {0,1,1,−1} {1,1,−1,0} {1,−1,0,−1} {1,−2,−1,0} {1,0,1,0} {0,0,1,1} {3,2,−1,−2} {1,0,−1,2} {0,2,−1,1} (which can, of course, easily be normalized. The system does not have a solution from $\{-1, 0, 1\}$).

(v) The smallest four-dimensional system with the smallest loop of size 2 is the following 19-10 one: `1234,4567,789A,ABCD,DEFG,GHI1,35CE,29BI,68FH,678J` shown in figure 3(*c*). It contains system (*a*) of figure 3 and it is the only MMP system with 19 vertices which has a solution from $\{-1, 0, 1\}$ for the corresponding vectors.

(vi) The two smallest four-dimensional systems with the smallest loops of size 2 that do not contain system (*a*) of figure 3 are the following 20-10 ones: `1234,4567,789A,ABCD,DEFG,GHI1,68FH,12JI,1J9B,345K,4KEC` and `1234,4567,789A,ABCD,DEFG,GHI1,68FH,2IAK,345J,4JEC,9ABK` shown in figures 4(*a*) and (*b*). The latter system is isomorphic to Kernaghan's system [29]. A solution to the former is `12...JK` = {0,0,0,1} {1,0,0,0} {0,1,1,0} {0,1,−1,0} {1,0,0,−1} {1,1,1,1} {1,−1,−1,1} {1, 1,−1,−1} {1,0,1,0} {0,1,0,1} {1,0,−1,0} {1,1,1,−1} {1,−1,1,1} {1,−1,−1,−1} {0,0,1,−1} {1,1,0,0} {1,−1,0,0} {0,0,1,0} {0,1,0,0} {1,0,0,1}.

(vii) All four-dimensional systems with up to 22 vectors and 12 edges with the smallest loops of size 2 which do have solutions from $\{-1, 0, 1\}$ contain at least one of the systems (*a*) and (*b*) of figure 4 and in many cases also (*a*) of figure 3. The two smallest four-dimensional systems with the smallest loops of size 2 that contain neither of the latter three systems are 22-13 systems (*c*) and (*d*) of figure 4: `1234, 4567, 789A, ABCD, DEFG, GHI1, 2ILA, 345J, 4JEC, 678K, 7KMG, 9ABL, FGHM` and `1234, 4567, 789A, ABCD, DEFG,`

[8] Still, they might be significant for other fields. For example, the two diagrams 19–13(5) given in section 3 (one of them is shown in figure 2(*b*) of [33] and the other in figure 2(*d*)) are equivalent to the Greechie diagrams with 19 *atoms* and 13 *blocks* and to our knowledge, the smallest Greechie diagram with three atoms per edge without 0–1 states known so far was the one given by Greechie [44, 45], with 27 atoms and 18 blocks. The system 38–19(5) from section 3 yields the smallest Greechie diagram with four atoms per block.

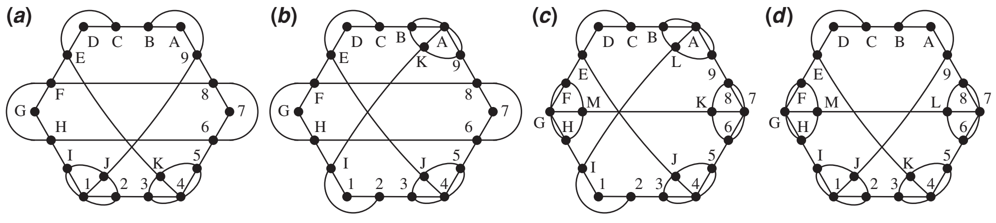

Figure 4. Smallest four-dimensional KS systems with loops of size 2: (1)—not containing system (*a*) of figure 3: (*a*) 20–11; (*b*) 20–11 isomorphic to Kernaghan [29]; (2)—containing neither system (*a*) of figure 3 nor systems (*a*) and (*b*) of this figure: (*c*) 22–13; (*d*) 22–13.

`GHI1, 12IJ, 345K, 678L, GML7, 1J9B, 4KEC, FGHM`. Their solutions are `12...M` = {1,1,0,0} {1,−1,0,0} {0,0,1,0} {0,0,0,1} {1,0,0,0} {0,1,1,0} {0,1,−1,0} {1,0,0,1} {1,−1,−1,−1} {1,1,1,−1} {1,−1,1,1} {1,0,−1,0} {0,1,0,1} {1,0,1,0} {1,1,−1,−1} {1,−1,−1,1} {1,−1,1,−1} {0,0,1,1} {0,1,0,0} {1,0,0,−1} {1,1,−1,1} {1,1,1,1} and `12... M` = {0,0,0,1} {1,0, 0,0} {0,1,1,0} {0,1,−1,0} {1,0,0,−1} {1,1,1,1} {1,−1,−1,1} {1,−1,1,− 1} {1,1,0,0} {0,0,1,1} {1,−1,0,0} {1,1,1,−1} {1,1,−1,1} {1,−1,−1,−1} {0,1, 0,−1} {1,0,1,0} {1,0,−1,0} {0,1,0,0} {0,0,1,0} {1,0,0,1} {1,1,−1,−1} {0,1, 0,1}.

(viii) As shown in [33], Peres' four-dimensional vectors [27] build a hexagon with 24 vertices and 24 edges and not with 22 edges as presented by Tkadlec in [39]. (One can easily verify that the edges {1, −1, 1, −1}{1, 1, −1, −1}{1, −1, −1, 1}{1, 1, 1, 1} and {1, −1, 1, 1} {1, −1, −1, −1}{1, 1, 1, −1}{1, 1, −1, 1} are missing in the middle of figure 1 in [39].) This Peres' four-dimensional 24-24 KS system contains systems (*a*) and (*c*) from figure 3 and all the systems from figure 4.

(ix) Four-dimensional systems with more than 41 vectors cannot have solutions from {−1, 0, 1}, and there are no such solutions to systems without `0–1` states with minimal loops of size 5 up to 41 vectors (there are altogether two such systems: 38-19(5) given in section 3 and a 40-20(5) system), which brings the Hasse (Greechie) diagram approach to the KS problem [38, 39] into question.

(x) It can easily be shown that a three-dimensional system of equations representing diagrams containing loops of sizes 3 and 4 cannot have a real solution. For loops of size 3, e.g. `123,345,561,` the proof runs as follows. Let us choose `1` = {1, 0, 0}, `2` = {0, 1, 0}, `3` = {0, 0, 1}, and `i` = $\{a_{i1}, a_{i2}, a_{i3}\}$, $i = 4, 5, 6$, and consider block `345`. Using `3·0` we get $a_{53} = 0$. Let us next consider group `561`. Using `5·1` = 0 we get $a_{51} = 0$ Hence, `5` = $\{0, a_{52}, 0\}$ and is therefore collinear with 2. Thus, the system cannot have a solution. The proof for loops of size 4 is similar, only a little longer.

(xi) The smallest three-dimensional systems without a `0–1` valuation have a minimal loop of size 5, 19 vertices and 13 edges (section 3, figure 2(*d*)), but they do not have real solutions. We scanned all systems with up to 30 vectors and 20 orthogonal triads and there are no KS vectors among them. This does not mean that Conway–Kochen's system (CK) [26, p 114] is the smallest KS system, though. It turns out that we cannot drop vectors that belong to only one edge from orthogonal triads because (a) there are cases where a solution to a full system allows `0–1` valuation while one to a system with dropped vectors does not and (b) there are cases where the full system does not allow `0–1` valuation but has no solution. So, CK is actually not a 31 but a 51 vector system: `123, 145,267, 2AB, 3CD, CEF, CGm, DIn, DKL, 6EM, 6KN, 7IO, 7GP, 4GQ, 4Ko, 5Ep, 5IS, ALW, AFX, BSY, BQZ, 3cf, 3de, cOh, dMT, cN9, dP8, eSl, fQg, iR1, jk1, iFa, jLb, kOU, kMV, RPH, RNJ` with 37 edges. (Tkadlec's claims [39]

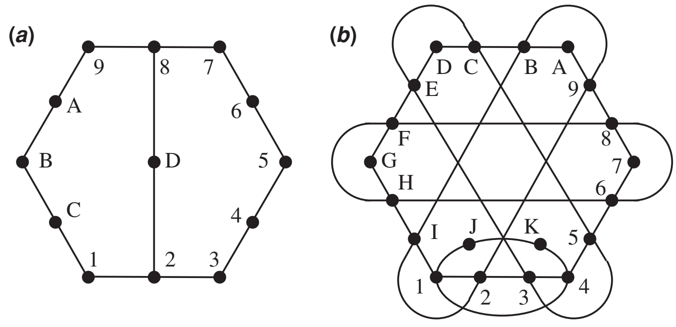

Figure 5. (*a*) System with dropped vectors that belong to only one edge (4,6,A,C,D) does not have any set of 0–1 states. Taking any of these vectors into account results in systems with at least one set of 0–1 states; (*b*) neither the system itself nor the systems obtained by dropping vectors (J or K or both) allow 0–1 states. The latter systems have solutions while the original system does not have any.

that CK can have 55 and 56 vectors and 54 edges are wrong.) A solution to CK is 12 · · · op = {0, 0, 1}{1, 0, 0}{0, 1, 0}{1, −1, 0}{1, 1, 0}{0, 1, −1}{0, 1, 1}{2, 5, 1}{2, 5, −1}{0, 1, 2}{0, 2, −1}{1, 0, 1}{1, 0, −1}{1, −1, −1}{1, 2, −1}{1, 1, −1}{2, −1, −5}{1, −1, 1}{2, −1, 5}{1, 1, 1}{1, −2, 1}{2, 1, 1}{2, −1, −1}{2, 1, −1}{2, −1, 1}{1, 1, 2}{1, 2, 0}{1, −1, −2}{2, −5, 1}{2, 1, 5}{2, 1, −5}{5, 2, −1}{5, −2, 1}{5, 1, 2}{5, −1, −2}{1, 2, 5}{1, −2, −5}{1, 0, 2}{1, 0, −2}{2, 0, 1}{2, 0, −1}{1, −5, 2}{2, −5, −1}{2, −1, 0}{2, 1, 0}{1, −2, 0}{1, 5, −2}{1, −2, −1}{1, 2, 1}{1, 1, −2}{1, −1, 2}. Thus, when all the vectors are taken into account, Bub's system [25] with 49 vectors and 36 edges: 123,345,167,AB6,AC4,DEG, DFH,F9O,E8V,5JI,7MN,GIa,HNh,7LT,5KR,DAe,UTS,PRS,1GP,3HU,3Vj,Pgh,Uba, 1Oi,VZg,OYb,6Xk,4Wn,Sde,dci,dfj,imn,jlk,akQ,hnQ,eQ2 is so far the smallest.

Let us see why we cannot drop vectors that belong to only one edge in detail. First, as mentioned above, if we drop all vectors that belong to only one edge, figure 5(*a*), we get: 123,35,567,789,9B,B1,28. This system has no 0–1 states. But if we add back a single such vector, say 123,345,567,789,9B,B1,28 or 123,35,567,789,9B,B1,2D8, the system has at least one set of 0–1 states. All these systems do have solutions. The opposite situation is given by figure 5(*b*). The system does not admit 0–1 states but has no solution. If we dropped J or K or both we would have a system with no 0–1 states and the systems would have solutions. For example, the system with dropped K has the following solution: 12 . . . IJ = {0, 0, 0, 1}{1, 0, 0, 0}{0, 1, 1, 0}{0, 1, −1, 0}{1, 0, 0, −1}{1, −1, −1, 1}{1, 1, 1, 1}{1, −1, 1, −1}{0, 1, 0, −1}{1, 0, −1, 0}{0, 1, 0, 1}{1, −1, 1, 1}{1, 1, 1, −1}{1, 1, −1, 1}{0, 0, 1, 1}{1, −1, 0, 0}{1, 1, 0, 0}{0, 0, 1, 0}{1, −1, −1, 0}. Second, in any KS diagram and therefore in Kochen–Specker (see figure 6), Bub, Peres and Conway–Kochen's ones in particular, only all vectors together make a complete description of their KS sets. Recall that one arrives from an n-tuple to an adjoining one by rotation around an $(n-2)$-dimensional subspace. For example, in figure 6 (*ii*) one starts with 123 and by rotations around 3, 5 and 7 one arrives at 789. So, 4 and 6 are indispensable for the construction and cannot be dropped as also shown by Larsson [28].

Special attention is deserved by the first KS graph ever, given by Kochen and Specker themselves [23]. We translated it into the MMP diagram notation in figure 6, where we also explain the correspondence between the two types of notation. Vectors of the KS graph are all contained in the three groups of five hexagons of the type shown in the inset (*ii*) of the figure. Since each such hexagon contains 13 vectors and since two hexagons in each group p, q and r

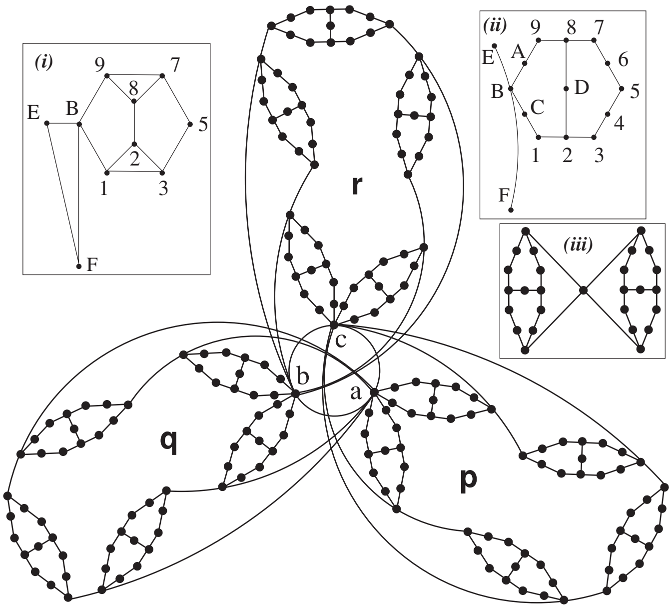

Figure 6. Historical Kochen–Specker 192 (117) graph [23, lemma 2, p 68, figure p 69] in the MMP diagram notation. Inset (*i*) shows the hexagon with the adjoining triangle from [23, lemma 1, p 68]. Inset (*ii*) shows the same graph in our MMP diagram notation: triangles translate as edges and everything else stays the same except that Kochen and Specker drop the vectors that do not share edges, in particular, vectors 4, 6, A, C and D. In [23, figure p 69] $a = p_0, b = q_0$ and $c = r_0$ hold. Here we glue these points together graphically. The groups of hexagons p, q and r here represent the hexagons containing p_i, q_i and r_i, $i = 0, \ldots, 4$ in [23, figure p 69]. Inset (*iii*) represents the 27(17)-point graph from [23, figure p 70] in the MMP diagram notation.

share a vector (vectors a, b and c, respectively) this makes 192 vectors (vertices) and 118 edges. By dropping vectors that do not share edges Kochen and Specker obtained 117 vectors.

Let us just mention here that Kochen and Specker's 27(17)-point graph (figure 6(*iii*)) provides a partial Boolean sub-algebra (characterizing the operations of commensurable observables in both quantum and classical mechanics) that cannot be embedded into a Boolean algebra (i.e., not all classical tautologies from the Boolean algebra correspond to equalities in the partial Boolean algebra). The graph does allow 0–1 states. It has properties similar to those of its hexagons (cf figure 5(*a*)) since it represents a vector system 123,345,567,789,9AB,BC1,4DA,EFG,GHI,IJK,KLM,MNO,OPE,HQN,1RK,7RE with the following components: $12 \cdots \mathrm{QR} = \{\{0, 1, -2\}\{5, 2, 1\}\{1, -2, -1\}\{1, 0, 1\}\{1, 1, -1\}\{2, -1, 1\}\{0, 1, 1\}\{2, 1, -1\}\{1, -1, 1\}\{1, 0, -1\}\{1, 2, 1\}\{5, -2, -1\}\{0, 1, 0\}\{0, 1, -1\}\{2, 1, 1\}\{1, -1, -1\}\{1, 1, 0\}\{1, -1, 2\}\{5, 1, -2\}\{0, 2, 1\}\{5, -1, 2\}\{1, 1, -2\}\{1, -1, 0\}\{1, 1, 1\}\{2, -1, -1\}\{0, 0, 1\}\{1, 0, 0\}\}$ which does have a set of 0–1 states when all vectors are taken into account and does not have it when the vectors that do not share edges are dropped. However, we obviously cannot dispense with vectors that build the system and therefore we cannot use this system for proving the KS theorem. Of course, Kochen and Specker were aware of this fact too and this is why they designed the aforementioned 15-hexagon 192-vector system.

(xii) The concept of *KS dual diagrams* [39, 46, 47] is apparently either a misnomer or insufficiently defined. Tkadlec claims that one arrives from a standard three-dimensional KS diagram to its dual so as to 'replace the role of points (vertices) and smooth curves (edges): points (vertices of the dual diagram) represent blocks (edges of the standard diagram) and maximal smooth blocks (maximal (?) edges of the dual diagram) represent atoms (vertices of the standard diagram)' [39]. The instructions for such a construction are ambiguous, but two figures are given in [39] and [46] and we have tested them. One is a dual diagram to Peres' 57 (33) diagram[9] and it reads `123, 345, 567, 869, 9AH, 8C2, 7DG, HG1, 4BA, CBD, 6gE, BhE, 3IJ, 2RO, 1VU, VPN, UML, JKN, OKL, IQM, RQP, jSK, jiQ, UWX, Veb, Gfa, HCZ, ZYb, XYa, WdC, edf, TFd, TcY, 1kE, 1lj, 1mT` and the other: `123,145,16C,768,7HK,4FB,GEC,89A,5IJ,HGI,EF9,KBD,JAD,CDV,KLM, BON, DgS, VUT,SP2,QRS,MQU,NPT,c6R,GPd,VWX,X3Y,3Ze,EZQ,abJ,YfA,cde,ehD, acW` is dual to Conway–Kochen's diagram. Of these two diagrams only the latter does not allow 0–1 states. The former has at least one set of 0–1 states. Then our solvers prove that the equation system corresponding to the latter diagram does not have a solution. Hence, neither of the two diagrams is a KS set, and we would expect a KS dual to be a KS set.

(xiii) We obtain the class of all remaining (non-KS) vectors from $\mathcal{H}^n$ by first filtering the MMP diagrams so as to keep only those that allow 0–1 states. Out of these, a second filter then keeps only those diagrams whose corresponding equations have solutions. Vectors corresponding to these solutions are the wanted non-KS vectors.

(xiv) The presented algorithms can easily be generalized beyond the KS theorem. One can use MMP diagrams to generate Hilbert lattice counterexamples, partial Boolean algebras and general quantum algebras which could eventually serve as an algebra for quantum computers [48]. One can also treat any condition imposed upon inner products in $\mathbb{R}^n$ to find solutions not by directly solving all nonlinear equations but also by first filtering the corresponding diagrams and solving only those equations that pass the filters.

Acknowledgments

One of us (MP) would like to acknowledge the support of the Ministry of Science, Education, and Sport of Croatia through the project *Quantum Theory of Information*, of the Computing Centre of the Faculty of Civil Engineering (in particular of A Karamatić who built the cluster under OpenMosix) and of the University Computing Centre of Zagreb through use of its clusters. He would also like to thank Jan-Åke Larsson from Linköping University, Sweden for drawing his attention to [28]. BM acknowledges support from the Australian Research Council.

References

[1] Cabello A and García-Alcaine G 1998 Proposed experimental tests of the Bell–Kochen–Specker theorem *Phys. Rev. Lett.* **80** 1797–9 (*Preprint* quant-ph/9709047)

[2] Simon C, Weinfurter H, Zukowski M and Zeilinger A 2000 Feasible Kochen–Specker experiment with single particles *Phys. Rev. Lett.* **85** 1783–6 (*Preprint* quant-ph/0009074)

[3] Meyer D A 1999 Finite precision measurement nullifies the Kochen–Specker theorem *Phys. Rev. Lett.* **83** 3751–4 (*Preprint* quant-ph/9905080)

[4] Kent A 1999 Noncontextual hidden variables and physical measurements *Phys. Rev. Lett.* **83** 3755–7 (*Preprint* quant-ph/9906006)

[5] Mermin N D 1999 A Kochen–Specker theorem for imprecisely specified measurement *Preprint* quant-ph/9912081

[6] Simon C, Brukner Č and Zeilinger A 2001 Hidden-variable theorems for real experiments *Phys. Rev. Lett.* **86** 4427–30 (*Preprint* quant-ph/0006043)

[7] Cabello A 2002 Finite precision measurement does not nullify the Kochen–Specker theorem *Phys. Rev.* A **65** 052101 (*Preprint* quant-ph/0104024)

[9] `123,39R,89A,47D,56E,DRE,EFG,CBD,NML,LKE,DJQ,QST,PJI,HKO,RVX,RUW,14Y,1Z5,4aA,5b8,8gB, AhF,7cH,6dI,CiO,GjP,7eM,6fS,ClN,GkT,NqX,PsV,OrU,MmU,SnV,HoX,IpW,TtW,2uB,2vF`.

[8] Huang Y-F, Li C-F, Yong-Sheng Zhang J-W P and Guo G-C 2002 Kochen–Specker theorem for finite precision spin-one measurement *Phys. Rev. Lett.* **88** 240402-1–4 (previous version) (*Preprint* quant-ph/0209038)
[9] Cabello A 2003 Kochen–Specker theorem for a single qubit using positive operator-valued measures *Phys. Rev. Lett.* **90** 190401-1–4 (*Preprint* quant-ph/0210082)
[10] Aravind P K 2003 Generalized Kochen–Specker theorem *Phys. Rev.* A **68** 051204 (*Preprint* quant-ph/0301074)
[11] Havlicek H, Krenn G, Summhammer J and Svozil K 2001 Colouring the rational quantum sphere and the Kochen–Specker theorem *J. Phys. A: Math. Gen.* **34** 3071–7 (*Preprint* quant-ph/9911040)
[12] Breuer T 2002 Kochen–Specker theorem for finite precision spin-one measurements *Phys. Rev. Lett.* **88** 24002 (*Preprint* quant-ph/0206035)
[13] Aravind P K and Lee-Elkin F 1998 Two noncolourable configurations in four dimensions illustrating the Kochen–Specker theorem *J. Phys. A: Math. Gen.* **31** 9829–34
[14] Gill R D and Keane M S 1996 A geometric proof of the Kochen–Specker no-go theorem *J. Phys. A: Math. Gen.* **29** L289–91 (*Preprint* quant-ph/0304013)
[15] Cabello A and García-Alcaine G 1996 Bell–Kochen–Specker theorem for any finite dimension $n \geqslant 3$ *J. Phys. A: Math. Gen.* **29** 1025–36
[16] Cabello A 2000 Kochen–Specker theorem and experimental tests on hidden variables *Int. J. Mod. Phys.* A **15** 2813–20 (*Preprint* quant-ph/9911022)
[17] Clifton R 2000 Complementarity between position and momentum as a consequence of Kochen–Specker arguments *Phys. Lett.* A **271** 1–7 (*Preprint* quant-ph/9912108)
[18] Isham C J and Butterfield J 1998 A topos perspective on the Kochen–Specker theorem: I. Quantum states as generalized valuations *Int. J. Theor. Phys.* **37** 2669–733 (*Preprint* quant-ph/9803055)
[19] Butterfield J and Isham C J 1999 A topos perspective on the Kochen–Specker theorem: II. Conceptual aspects and classical analogues *Int. J. Theor. Phys.* **38** 827–59 (*Preprint* quant-ph/9808067)
[20] Hamilton J, Isham C J and Butterfield J 2000 A topos perspective on the Kochen–Specker theorem: III. Von Neumann algebras as the base category *Int. J. Theor. Phys.* **39** 1413–36 (*Preprint* quant-ph/0009039)
[21] Butterfield J and Isham C J 2002 A topos perspective on the Kochen–Specker theorem: IV. Interval valuations *Int. J. Theor. Phys.* **41** 613–39 (*Preprint* quant-ph/0107123)
[22] Barrett J and Kent A 2004 Noncontextuality, finite precision measurement and the Kochen–Specker *Stud. Hist. Phil. Mod. Phys.* **35** 151–76 (*Preprint* quant-ph/0309017)
[23] Kochen S and Specker E P 1967 The problem of hidden variables in quantum mechanics *J. Math. Mech.* **17** 59–87
[24] Zimba J and Penrose R 1994 On Bell non-locality without probabilities: more curious geometry *Stud. Hist. Phil. Sci.* **24** 697–720
[25] Bub J 1996 Schütte's tautology and the Kochen–Specker theorem *Found. Phys.* **26** 787–806
[26] Peres A 1993 *Quantum Theory: Concepts and Methods* (Dordrecht: Kluwer)
[27] Peres A 1991 Two simple proofs of the Bell–Kochen–Specker theorem *J. Phys. A: Math. Gen.* **24** L175–8
[28] Larsson J-Å 2002 A Kochen–Specker inequality *Europhys. Lett.* **58** 799–805 (*Preprint* quant-ph/0006134)
[29] Kernaghan M 1994 Bell–Kochen–Specker theorem for 20 vectors *J. Phys. A: Math. Gen.* **27** L829–30
[30] Cabello A, Estebaranz J M and García-Alcaine G 1996 Bell–Kochen–Specker theorem: a proof with 18 vectors *Phys. Lett.* A **212** 183–7 (*Preprint* quant-ph/9706009)
[31] Pavičić M 2002 Quantum computers, discrete space, and entanglement *SCI 2002/ISAS 2002 Proc. 6th World Multiconference on Systemics, Cybernetics, and Informatics* (*SCI in Physics, Astronomy and Chemistry*, vol 17) ed N Callaos, Y He and J A Perez–Peraza (Orlando, FL: SCI) pp 65–70 (*Preprint* quant-ph/0207003)
[32] Pavičić M 2003 Constructing quantum reality (invited talk) *Proc. The Role of Mathematics in Physical Sciences,* (Veli Losing, Croatia August 2003) (to appear)
[33] Pavičić M 2004 Kochen–Specker algorithms for qubits *Proc. Quantum Communication Measurement and Computing: The Seventh Int. Conf. on Quantum Communication, Measurement and Computing (Glasgow, July 2004)* ed S M Barnett, E Andersson, J Jeffers, P Őhberg and O Hirota (New York: AIP) pp 195–8
[34] Smith D 2004 Orthomodular Bell–Kochen–Specker theorem *Int. J. Theor. Phys.* **43** 2023–7
[35] Svozil K and Tkadlec J 1996 Greechie diagrams, nonexistence of measures and Kochen–Specker-type constructions *J. Math. Phys.* **37** 5380–401
[36] Tkadlec J 2000 Diagrams of Kochen–Specker constructions *Int. J. Theor. Phys.* **39** 921–6
[37] Hultgren III B O and Shimony A 1977 The lattice of verifiable propositions of the spin–1 system *J. Math. Phys.* **18** 381–94
[38] McKay B D, Megill N D and Pavičić M 2000 Algorithms for Greechie diagrams *Int. J. Theor. Phys.* **39** 2381–406 (*Preprint* quant-ph/0009039)
[39] McKay B D 1998 Isomorph-free exhaustive generation *J. Algorithms* **26** 306–24

[40] McKay B D 1990 `nauty` User's Guide (version 1.5) *Computer Science Department, Australian National University Report* TR-CS-90-02

[41] Pavičić M, Merlet J-P, McKay B D and Megill N D 2004 Exhaustive enumeration of Kochen-Specker vector systems *The French National Institute for Research in Computer Science and Control Research Reports* RR-5388 *Preprint* http://www.inria.fr/rrrt/rr-5388.html

[42] Duffin K L and Barrett W A 1994 Spiders: a new user interface for rotation and visualization of n-dimensional points sets *Proc. 1994 IEEE Conf. on Scientific Visualization* ed D Bergeron and A Kaufman (Los Alamitos, CA: IEEE Computer Society Press) pp 205–11

[43] Ritt J F 1950 *Differential Algebra Coll. Publ. 33* (New York: American Mathematical Society)

[44] Greechie R J 1971 Orthomodular lattices admitting no states *J. Comb. Theor.* **10** 119–32

[45] Pták P and Pulmannová S 1991 *Orthomodular Structures as Quantum Logics* (Dordrecht: Kluwer)

[46] Tkadlec J 2001 Representations of orthomodular structures *Ordered Algebraic Structures: Nanjing, Proc. Nanjing Conf.* (*Algebra, Logic and Applications Series* vol 16) ed W Charles Holland (London: Taylor and Francis) pp 153–8

[47] Svozil K 2000 On generalized probabilities: correlation polytopes for automaton logic and generalized urn models, extensions of quantum mechanics, and parameter cheats *Preprint* quant-ph/0012066

[48] Megill N D and Pavičić M 2000 Equations, states, and lattices of infinite-dimensional Hilbert space *Int. J. Theor. Phys.* **39** 2337–79 (*Preprint* quant-ph/0009038)