Recognizing large-degree Intersection Graphs of Linear 3-Uniform Hypergraphs

Yury Metelsky[1], Erich Prisner[2], Stanislav Suzdal[1], and Regina Tyshkevich[1]

1: *DEPARTMENT OF MECHANICS AND MATHEMATICS*
BELARUS STATE UNIVERSITY
220050 MINSK, BELARUS
E-MAIL: regina@mmf.bsu.unibel.by

2: *MATHEMATISCHES SEMINAR*
UNIVERSITÄT HAMBURG
20256 HAMBURG, GERMANY
E-MAIL: prisner@math.uni-hamburg.de

1 Introduction

The *intersection graph* or the *line graph* $\Omega(H)$ of a hypergraph H is defined as follows:

1) the vertices of $\Omega(H)$ are in a bijective correspondence with the edges of H,

2) two vertices are adjacent in $\Omega(H)$ if and only if the corresponding edges intersect.

Characterizing and recognizing intersection graphs of hypergraphs with some additional property P is one of the central problems in intersection graph theory.

It is well known that if the property P is closed with respect to deleting edges, then the class of intersection graphs of members in P is closed under induced subgraphs. Then the class can be characterized by means of a list F of forbidden induced subgraphs. If F is finite, then it is called a *finite FIS-characterization*. Obviously, the recognition problem can be solved in polynomial time provided the class $\Omega(P)$ has a finite FIS-characterization.

A hypergraph is called *k-uniform* if all its edges have the same cardinality k. In a *linear* hypergraph, no two edges have two vertices in common. We define L_k and L_k^l as the classes of intersection graphs of k-uniform hypergraphs, and of linear k-uniform hypergraphs, respectively.

The classes L_2 and L_2^l (the line graphs of multigraphs and of simple graphs, respectively) have been studied for a long time. Finite FIS-characterizations ([1], [3]) are obtained, and efficient algorithms for recognizing these classes are known ([4], [8], [14], [16]).

The situation changes qualitatively after taking $k = 3$ instead of $k = 2$. Lovász posed the problem of characterizing the class L_3, and noted that no finite FIS-characterization exists here [10]. It is proven in [5] that recognizing intersection graphs of linear k-uniform hypergraphs is NP-complete for $k \geq 3$. In [2], for an arbitrary constant k, a characterization of L_k in terms of clique coverings which is a generalization of the well known Krausz theorem [7] characterizing L_2^l is given. Such so-called *Krausz characterizations* are useful, but do not solve the recognition problem, since here one must look over all clique coverings of an arbitrary graph G.

[1]The first, third and fourth author were supported by INTAS and the Belarus Government, Project INTAS-BELARUS 97-0093.

[2]The second author gratefully acknowledges support by the 'Wilhelm-Blaschke-Gedächtnis-Stiftung', and by FONDAP (Chile) during June/Juli 98 at the Universidad de Chile, where part of the research has been done.

Actually recognizing graphs in L_k is NP-complete for $k \geq 4$ [15]. The question whether or not members of L_3 can be recognized efficiently is still open, but recognizing intersection graphs of 3-uniform *simple* hypergraphs is NP-complete as well [15].

In 1980 R.N. Naik, S.B. Rao, S.S. Shrikhande, and N.M. Singhi proved the existence of a finite FIS-characterization of graphs in L_3^l with bound on minimum vertex degree $\delta(G) \geq 75$ [12]. In 1982 the same authors reduced their bound to $\delta(G) \geq 69$ [13]. In [11] an algorithm recognizing $G \in L_3^l$ in polynomial time under the condition $\delta(G) \geq 19$ is proposed and a finite FIS-characterization of the class of graphs handled is obtained. The same recognition problem is independently solved in [6].

An algorithm solving this problem under the condition $\delta(G) \geq 13$ is proposed in this paper. The complexity of the algorithm is $\mathcal{O}(nm)$, where n and m are the number of vertices and edges, respectively. Whether or not a finite FIS-characterization of the graphs considered exists is still open.

Similar results for $k \geq 4$ cannot be expected, since the following two theorems hold.

Theorem [11]. *For $k \geq 4$ and an arbitrary constant δ, the set of all graphs G in L_k^l with minimum degree at least δ has no finite FIS-characterization.*

Theorem [5]. *For any δ, the problem to test membership in L_4^l is NP-complete for graphs of minimum degree δ.*

2 Main Tools

All graphs considered are finite, undirected and have no loops and multiple edges.

The vertex set of a graph G is denoted by VG. If $N(v) = N_G(v)$ is the neighborhood of a vertex v in G, then $N[v] = N(v) \cup \{v\}$, $deg(v) = |N(v)|$ is the *degree* of v, $\delta(G)$ is the minimum vertex degree of G.

An arbitrary subset of pairwise adjacent vertices in a graph is called a *clique*. A *maximal* clique (or simply *maxclique*) is maximal with respect to inclusion. For a positive integer k, a *k-clique* (or *$\geq k$-clique*) is a clique with exactly (at least) k vertices, and *k-maxcliques* and *$\geq k$-maxcliques* are defined analogously.

A family $Q = \{C_1, C_2, \ldots, C_q\}$ of cliques of a graph G is called a *Krausz k-partition* with *clusters C_i* if the following conditions hold:

1) every edge of G belongs to exactly one cluster C_i,

2) every vertex of G belongs to at most k clusters of Q.

A *fragment* is any subfamily of some Krausz k-partition. A vertex is *covered* by a fragment if it is contained in some cluster of the fragment. An edge is *covered* by a fragment if both of its vertices belong to the same cluster in the fragment. A *Krausz-extension* of a fragment is any Krausz k-partition containing the fragment.

The algorithm below uses any of the algorithms for recognizing line graphs ([4], [8], [16]) and is based on the following two facts.

Theorem 1 [2]. *A graph belongs to the class L_k^l if and only if there exists a Krausz k-partition of the graph.*

Lemma 2 ([13], [9], [5], [6]). *Each $\geq (k^2-k+2)$-maxclique of a graph is a cluster of any its Krausz k-partition.*

From now on we only consider the case $k=3$. We omit k and write "Krausz partition". We call a ≥ 8-clique *large*, and a 6- and 7-clique *prelarge*.

We assume throughout Lemma 3 to Lemma 10 that $G \in L_3^l$, F is a fragment in G and Q is a Krausz-extension of F. In Lemma 6 the case $F=\emptyset$ is not excluded, Q is an arbitrary Krausz-partition here.

Lemma 3 . *If a vertex v is covered by exactly two clusters C_1 and C_2 in F and $C = N(v) \setminus (C_1 \cup C_2) \neq \emptyset$, then $C \cup \{v\}$ is a cluster in Q.* □

Lemma 4 . *Let C_i be a cluster in F, $|C_i| \geq 4$, C be a clique in G and $C \not\subseteq C_i$. Then $|C \cap C_i| \leq 3$. In particular C_i is a maxclique.*

Proof: If x is a vertex in $C \setminus C_i$, then all edges xy with $y \in C_i \cap C$ must be covered by different clusters, whence there are at most three of them. □

We say that a clique C *touches* a clique D (*in a vertex* v) if $C \cap D = \{v\}$. A clique *touches* a fragment F if the clique touches some cluster in F.

Put $H = G - EF$ where EF is the set of edges covered by F. In what follows $N_k[a] = N_k^H[a]$ is the ball of radius k with center a in H,

$$N_k(a) = N_k[a] \setminus \{a\}, \quad N_k\{a\} = N_k[a] \setminus N_{k-1}[a].$$

Lemma 5 . *Every prelarge maxclique in H touching F is a cluster in Q.*

Proof: Let C be a maxclique in H, $C_1 \in F$ and $C_1 \cap C = \{v\}$. Then

$$C \subseteq C_2 \cup C_3,\ C_2, C_3 \in Q,\ C_2 \cap C_3 = \{v\}.$$

Suppose $C \neq C_1$ and $C \neq C_2$. Then $C \not\subseteq C_i, i = 1,2$. Lemma 4 implies $|C \cap C_1|, |C \cap C_2| \leq 3$ and therefore $|C| \leq 5$. □

Lemma 6 . *Let C be a maxclique in H, and $a \in VH \setminus C$. Furthermore, let C contain a vertex adjacent to a and at least five vertices not adjacent to a. Then $C \in Q \setminus F$.*

Proof: Assume that C is not contained in a cluster in $Q \setminus F$ and let $b \in C$ be a vertex adjacent to a. Then $C \subseteq N_1[b]$ and $N_1[b]$ is the union of at most three clusters in $Q \setminus F$ containing b. Let C_1 be one of them, $a \in C_1$. So five vertices not adjacent to a must be contained in the union of two other clusters. The vertex b belongs to both of them, a contradiction with Lemma 4. Then the maximality of the ≥ 6-clique C and Lemma 4 prove the statement. □

Remark. Lemma 6 is interesting in the case when C is a prelarge clique. Then the conditions in the lemma require

We call a clique C containing a non-covered vertex b b-*good*, and the vertex b *good* if one of the conditions holds:

1. C is a large clique of H,

2. C is a prelarge clique of H touching F,

3. C is such as in Lemma 6 (Remark).

Lemma 7 . *Let a and b be adjacent vertices of H, and a be covered by F, b be non-covered by F. If $deg_G(b) \geq 15$, then b is a good vertex.*

Proof: Without loss of generality assume that b neither belongs to a large clique in H nor to a prelarge clique in H touching F. The fragment $Q \setminus F$ contains a cluster C_1 covering both a and b. Since C_1 touches F, then $|C_1| \leq 5$. But $deg(b) \geq 15$ so there exists a cluster C_2 in Q such that $b \in C_2$, $|C_2| = 7$. If $N_1(a) \cap C_2 \neq \{b\} = C_1 \cap C_2$, then there exists a cluster C_3 in Q containing a. One has $|C_2 \cap C_3| \leq 1$. So $|C_2 \cap N_1(a)| \leq 2$, C_2 is b-good, b is good. □

We say that an (F, a)-*condition holds* if, for some vertex a, the following is valid:

1. $\delta(G) \geq 13$,

2. a is covered by F,

3. there exists a non-covered neighbor of a, but none of them is good.

A pair (C_1, C_2) of maxcliques in H is called an (F, a)-*pair* if it satisfies the conditions:

1. $|C_1 \cap C_2| = \{a\}$,

2. $C_1 \cup C_2 = N_1[a]$,

3. $|C_i| = 4$ or 5, $i = 1, 2$.

For an arbitrary (F, a)-pair $P = (C_1, C_2)$, a t-tuple $T = (C_1, C_2, C_3, \ldots, C_t)$ of maxcliques C_i in H is called an (F, a)-*tuple associated with* P (or simply (F, a)-*tuple*) if it satisfies the following conditions 1–6.

1. $5 \leq |C_i| \leq 6,\ i = 3, \ldots, t$.

2. Each vertex $v \in N_1(a)$ belongs to exactly three cliques of T.

3. If a vertex v belongs to exactly two cliques C_i and C_j, $i \neq j$, then $D_v = (N_1(v) \setminus (C_i \cup C_j)) \cup \{v\}$ is a clique.

   Adding each clique D_v with more than one vertex to T we obtain the *closure* $\overline{T}$ of T.

4. Each two cliques of $\overline{T}$ have at most one common vertex.

5. No vertex $v \in VG$ belongs to more than three cliques of $\overline{T}$.

6. If v belongs to three cliques of $\overline{T}$, then $N_1[v]$ is precisely the union of these cliques.

Lemma 8 . *Let an (F,a)-condition hold. Then*

1) $13 \leq deg(b) \leq 14$ for every non-covered neighbor b of a,

2) there is an (F,a)-pair contained in $Q \setminus F$.

Proof: Statement 1) follows by Lemma 7 and $\delta(G) \geq 13$. For 2) consider any noncovered neighbor b of a. The edge ab is contained in some cluster $C_1 \in Q \setminus F$. Since b is not good, $|C_1| \leq 5$. We claim that b belongs to no 7-cluster. Otherwise, by Lemma 6, this clique would contain at least three neighbors b, b_2, b_3 of a. The edges ab, ab_2, ab_3 would have to be in different clusters, a contradiction to a being in just three of them. By Lemma 7, $deg(b) = 13$ or 14. Hence $|C_1| \geq 4$ and there exist two more clusters

$$C_3, C_4 \in Q \setminus F,\ b \in C_i,\ i = 3,4,\ |C_3| = 6,\ |N_1(a) \cap C_3| \geq 2.$$

No vertex in C_3 is covered by F, for otherwise b would be good. Let $C_3 \cap N_1(a) \supseteq \{w, b\}$. Then there exists the fourth cluster $C_2 \in Q \setminus F$ containing a, w. The vertex w is adjacent to a and non-covered by F, so $deg(w) = 13$ or 14, and $|C_2| = 4$ or 5. Obviously, (C_1, C_2) is an (F,a)-pair.
□

Lemma 9 . *Let an (F,a)-condition hold and H have at least two (F,a)-pairs. Then*

1) $N_1(a)$ has no vertex covered by F;

2) $13 \leq deg(v) \leq 14$ for any $v \in N_1(a)$;

3) H has an (F,a)-tuple $T = (C_1, C_2, \ldots)$ contained in $Q \setminus F$;

4) for any (F,a)-tuple $T' = (C'_1, C'_2, \ldots)$ different from T, the pairs (C_1, C_2), (C'_1, C'_2) are different;

5) if H has more than one (F,a)-tuples, then every vertex $\alpha \in N_2\{a\}$ belongs to at least two cliques in each (F,a)-tuple.

Proof: We have $6 \leq |N_1(a)| \leq 8$. Let $P = (C_1, C_2),\ P' = (C'_1, C'_2)$ be different (F,a)-pairs, and by Lemma 8 we may assume $P \subseteq Q \setminus F$. Further distinguish the cases $|N_1(a)| = 6, 7, 8$.

I. $|N_1(a)| = 6$. Without loss of generality assume that

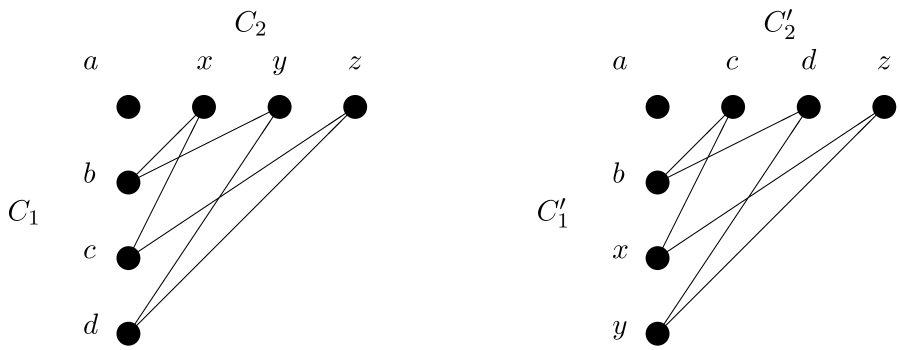
$$C_1 = \{a,b,c,d\},\ C_2 = \{a,x,y,z\},\ C'_1 = \{a,b,x,y\},\ C'_2 = \{a,c,d,z\}. \tag{1}$$

Since $F \cup P \subseteq Q$ and since C'_1, C'_2 lie in H, H has the edges bx, by, cz and dz, and no two of them belong to the same cluster. So we have distinct clusters $C_{bx}, C_{by}, C_{cz}, C_{dz} \in Q \setminus F$, where here and in the following C_{uv} denotes the cluster containing $u, v \in N_1(a)$. The vertices b and z are contained in three clusters in $Q \setminus F$ and therefore they are not covered by F. Hence $deg(b) = deg(z) = 13$ and $|C_{uv}| = 6$ for any of the four clusters mentioned above. By Lemma 5 no $v \in N_H(a)$ is covered by F. As in the proof of Lemma 8, c cannot lie in a 7-cluster, thus $deg(c) = deg(d) = 13$. Using Lemma 6 there exist two more 6-clusters of the forms C_{cu} and C_{dv} with $u \neq v$, say C_{cx} and C_{dy}. Obviously, the 8-tuple

$$T = (C_1, C_2, C_{bx}, C_{by}, C_{cz}, C_{dz}, C_{cx}, C_{dy}) \tag{2}$$

is an (F, a)-tuple contained in $Q \setminus F$.

The bipartite graph induced by the edges of H mentioned above is shown in Figure 1,a.



a b

Figure 1

II. $|N_1(a)| = 7$. Arguing as above, assume

$$C_1 = \{a, b, c, d, e\},\ C_2 = \{a, x, y, z\},\ C'_1 = \{a, b, c, x, y\},\ C'_2 = \{a, d, e, z\}. \tag{3}$$

We obtain the bipartite graph shown in Figure 2,a and the 8-tuple

$$(C_1, C_2, C_{bx}, C_{by}, C_{cx}, C_{cy}, C_{dz}, C_{ez})$$

contained in $Q \setminus F$. We have $|C_{uv}| = 6$ for each $u, v \in N_1(a)$; $deg(b) = deg(c) = 14,\ deg(z) = 13$; no vertex in $N_1(a)$ is covered by F. Further $13 \leq deg(d) \leq 14$ since d is not good. Hence there is one more cluster C_d containing d; $|C_d| < 6$ since d is not good and $|N_1(a) \cap C_d| = 1$. So $|C_d| = 5$. Analogously for e, $|C_e| = 5$.

Thus,

$$T = (C_1, C_2, C_{bx}, C_{by}, C_{cx}, C_{cy}, C_{dz}, C_{ez}, C_d, C_e) \tag{4}$$

is an (F, a)-tuple contained in $Q \setminus F$.

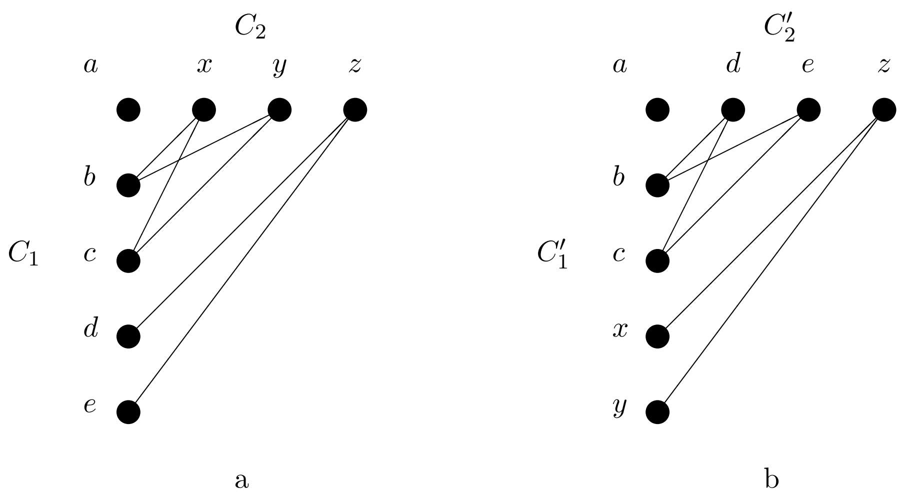


Figure 2

III. $|N_1(a)| = 8$. Arguing as above, put

$$C_1 = \{a, b, c, d, e\},\ C_2 = \{a, w, x, y, z\},\ C_1' = \{a, b, c, w, x\},\ C_2' = \{a, d, e, y, z\}, \tag{5}$$

obtain the bipartite graph shown in Figure 3,a and the (F, a)-tuple

$$T = (C_1, C_2, C_{bw}, C_{bx}, C_{cw}, C_{cx}, C_{dy}, C_{dz}, C_{ey}, C_{ez}) \tag{6}$$

contained in $Q \setminus F$. We have

$$5 \leq |C_{uv}| \leq 6,\ 13 \leq deg(v) \leq 14$$

for $v \in N_1(a)$.

The statements 1)–3) are proved for any case.

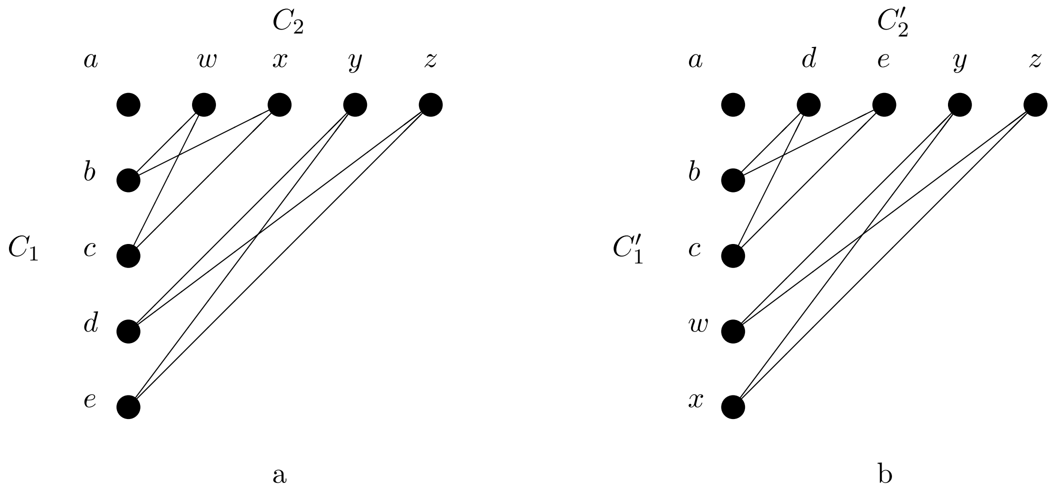


Figure 3

4) Let there exist another (F, a)-tuple $T'(\neq T)$ associated with the pair $P = (C_1, C_2)$. It is proved above that fixed different pairs P and P' uniquely (up to notation) define the lower indices of cliques in T, i.e. the vertices in $N_1(a)$ contained in these cliques. So the graphs shown in Figures 1,a–3,a are defined. Thus, in the case I, for example, T' has the form

$$T' = (C_1, C_2, C'_{bx}, C'_{by}, C'_{cz}, C'_{dz}, C'_{cx}, C'_{dy}).$$

The 6-clique C'_{bx} touches the cluster C_1 in vertex b, and then is contained in $Q \setminus F$ as C_{bx}. Both the cliques contain the edge bx and therefore coincide. The same holds for all other cliques, hence $T' = T$. Analogously for the case III. In the case II $C'_{uv} = C_{uv}$, and each of the cliques C'_d, C_d complements $(C_1 \cup C'_{dz}) = (C_1 \cup C_{dz})$ to $N_1[d]$ and, consequently, $C'_d = C_d$. Analogously for C_e and C'_e.

5) Let T' be an (F, a)-tuple different from T and associated with the pair $P' = (C'_1, C'_2)$. Then the pairs P, P' are different and one can assume that C'_1, C'_2 are defined by one of the equalities (1), (3), (5). Then T has the form (2), (4), (6), respectively. We have

$$T' = (C'_1, C'_2, C'_{bc}, C'_{bd}, C'_{xc}, C'_{xz}, C'_{yd}, C'_{yz}),$$

$$T' = (C'_1, C'_2, C'_{bd}, C'_{be}, C'_{cd}, C'_{ce}, C'_{xz}, C'_{yz}, C'_x, C'_y),$$

$$T' = (C'_1, C'_2, C'_{bd}, C'_{be}, C'_{cd}, C'_{ce}, C'_{wy}, C'_{wz}, C'_{xy}, C'_{xz}),$$

respectively. Further the necessary statement is verified directly. Let, for example, in the case II $\alpha \in C_d$. Hence $\alpha \in C'_{bd} \cup C'_{cd}$ (see Fig. 2,b). Let, for example, $\alpha \in C'_{bd}$. Then $\alpha \in C_{bx} \cup C_{by}$. The same for T'. $\square$

Lemma 10 . *If an (F,a)-condition holds, then the set $F \cup T'$ is a fragment for every (F,a)-tuple T'.*

Proof: Let T be an (F,a)-tuple constructed in Lemma 9, i.e. contained in $Q \setminus F$, and T' be an arbitrary (F,a)-tuple different from T. Put $R = Q \setminus (F \cup \overline{T})$ and prove that $Q' = F \cup \overline{T'} \cup R$ is a Krausz-partition of G. The ball $N_2[a]$ coincides with the union of all cliques in T' as well as with the union of all clusters in T. The analogous statement is true with respect to $N_3[a]$, $\overline{T'}$ and $\overline{T}$. Therefore only the vertices of $N_3\{a\}$ can belong to the cliques both in $\overline{T'}$ and R.

Thus, a nontrivial situation occurs only in these vertices. Let $v \in N_3\{a\}$. If v belongs to three cliques $D_1,\ D_2,\ D_3$ of $\overline{T}$ (or $\overline{T'}$), then $N_1[v] \setminus (D_1 \cup D_2 \cup D_3) = \emptyset$, so v belongs to no clique of R. The conditions in definition of Krausz-partition could be violated only if v belongs to exactly one clique D in $\overline{T}$ and two cliques D_1' and D_2' in $\overline{T'}$. In this case, clearly, D coincides with the set $N_1[v] \cap N_3[a]$. Let ET and $H(ET)$ be the set of edges contained in cliques of T and the graph induced by ET, respectively. Denote by $det_T(\alpha)$ the degree of α in the graph $H(ET)$.

If one more vertex u belongs both to $N_3\{a\}$ and D, then the pair $\{u,v\}$ must be contained both in D_1' and D_2'. So v is the unique vertex contained both in $N_3\{a\}$ and D.

Let

$$D = \{v, \alpha_1, \ldots, \alpha_p, \ldots, \alpha_k\},\ D_1' = \{v, \alpha_1, \ldots, \alpha_p\},\ D_2' = \{v, \alpha_{p+1}, \ldots, \alpha_k\}.$$

The definition of $\overline{T}$ and $\overline{T'}$ implies

$$|D| = deg_{H-ET}(\alpha_1) + 1,\ |D_1'| = deg_{H-ET'}(\alpha_1) + 1,\ |D_2'| = deg_{H-ET'}(\alpha_k) + 1.$$

We have $|D| > |D_1'|$, consequently, $deg_T(\alpha_1) < deg_{T'}(\alpha_1)$. By Lemma 9, α_1 belongs (in the situation considered) to exactly two cliques both in T and T' of orders 5 or 6. Therefore

$$deg_{T'}(\alpha_1) - deg_T(\alpha_1) \le 2 \cdot 5 - 2 \cdot 4 = 2.$$

Further we have

$$|D_2'| = |D| - |D_1'| + 1 = deg_{H-ET}(\alpha_1) - deg_{H-ET'}(\alpha_1) + 1 = deg_{T'}(\alpha_1) - deg_T(\alpha_1) + 1 \le 3.$$

So $deg_{H-ET'}(\alpha_k) = |D_2| - 1 \le 2$. Finally, since α_k is not covered by F,

$$deg_G(\alpha_k) = deg_H(\alpha_k) = deg_{T'}(\alpha_k) + deg_{H-ET'}(\alpha_k) \le 2 \cdot 5 + 2 = 12.$$

□

3 Algorithm

Before giving the formal algorithm, let us outline the idea of our approach.

Assume we have some nonempty set F of cliques, which is a fragment if and only if $G \in L_3^l$. For example, by Lemma 2, all large maxcliques must be members of all Krausz-partitions, thus an arbitrary set of large maxcliques can stand for F.

If every vertex is covered by some element of F, we are almost done, as has been shown in [11], [6]. Things become slightly easier if we assume that every vertex is covered by 1 or 3 elements of F, but this can be achieved by Lemma 3. Now we delete all edges covered by members of F, and the resulting graph is a line graph if and only if $G \in L_3^l$. Note that the large maxcliques cover the vertex set of graphs in L_3^l of minimum degree at least 19.

For graphs in L_3^l of smaller minimum degree, the large maxcliques do not have to cover the vertex set. Anyway we start with a nonempty set F of cliques in G. Then the algorithm extends F adding one or more cliques on each step. Besides the following two conditions are quaranteed if $G \in L_3^l$:

- the set F_{k+1} obtained on step $k+1$ is a fragment if and only if F_k is a fragment,
- if F_k is a fragment and G has a vertex noncovered by F_k, then one can fulfil step $k+1$.

Thus the algorithm constructs a fragment F' covering all vertices if and only if F is a fragment. Moreover, each vertex belongs to 1 or 3 clusters in F'.

So it remains to choose an initial set F. If there is a large maxclique C in G, then we put $F = \{C\}$. Otherwise there is a prelarge maxclique C containing some fixed vertex z which is included to F. In the latter case, if the algorithm fails to construct an appropriate fragment F' or the graph $G - EF'$ is not a line graph, then G has no Krausz-partition with C as a cluster. Therefore take as C another prelarge maxclique containing z. At worst one must look over all prelarge maxcliques containing z.

Algorithm 11 .
Instance: A connected graph G with $\delta(G) \geq 13$.
Question: Is G the intersection graph of some linear 3-uniform hypergraph?

`During the algorithm every vertex` v `of a current graph possesses some non-negative integer weight` $w(v)$ `and can be labelled. Let` H `and` F `be a current graph and a current fragment, respectively.`

Routine

(1) `Find and fix a vertex` z `of maximum degree in` G.

(2) `Check whether` G `contains some large maxclique as follows:`

  (2.1) `For any` $S \subseteq N_G(z)$, $|S| = 19$, S `must contain a 7-clique` K. `Extend` $K \cup \{z\}$ `to a maximal clique` C.

  (2.2) `If` $deg_G(z) \leq 18$, `then we check within the neighborhood of each vertex in` VG.

(3) `Initialisation:`

  (3.1) `If` G `has a large maxclique` C, `then we start with` $F = \{C\}$.

  (3.2) `If` G `has no large maxclique, then there is a prelarge maxclique` C `containing` z. `Put` $F = \{C\}$.

(3.3) $H := G - EF$,

(4) While there exists a vertex v with $w(v) = 0$ do:

(4.1) Fix a *leading* vertex a which is a non-labelled vertex with $w(v) > 0$.

(4.2) Find a good vertex in $N_1(a)$ as follows:

(4.2.1) Find a vertex b with $w(b) = 0$ in $N_1(a)$. If there exists no such vertex, then label a and go to (4.1).

(4.2.2) Check whether b is good as follows:

(a) For any $S \subseteq N_G(b)$, $|S| = 19$, S must contain a 7-clique and, consequently, b is good.

(b) If $|N_1(b)| \leq 18$, then b is good if and only if there exists a b-good clique in $N_1[b]$.

(4.2.3) If b is not good, then choose another vertex in (4.2.1).

(4.3) If there exists a good $b \in N_1(a)$, then find a b-good clique C_b. Put $F := F \cup \{C_b\}$. Perform Subroutine(C_b).

(4.4) If there is no good vertex in $N_1(a)$, then the conditions $6 \leq |N_1(a)| \leq 8$ and, for any $b \in N_1(a)$, $|N_1(b)| \leq 14$ must hold. There must exist an (F, a)-pair $P = (C_1, C_2)$ in $N_1[a]$.

(4.5) If P is a unique (F, a)-pair in $N_1[a]$, then put $F := F \cup P$. Perform Subroutine(C_1, C_2).

(4.6) If there are at least two (F, a)-pairs, then there exist an (F, a)-tuple $T = (C_1, \dots, C_t)$ in $N_2[a]$. Put $F := F \cup T$. Perform Subroutine($C_1, \dots, C_t$).

(5) The resulting graph H must have some Krausz 2-partition J. $F \cup J$ is a Krausz 3-partition of G.

Subroutine($C_1, C_2, \dots, C_p$).

(S.1) $H := H - EF$.

(S.2) For any $v \in \bigcup_{i=1}^{p} C_i$, perform $w(v) := w(v) + |\{C_i : v \in C_i, i = 1, 2, \dots, p\}|$.

(S.3) For any $v \in VG$, the condition $w(v) \leq 3$ must hold.

(S.4) For any $v \in VG$, if $w(v) = 3$, then the condition $N_1(v) = \emptyset$ must hold.

(S.5) Fix a vertex $u \in VG$ with $w(u) = 2$. The set $N_1[u]$ must be a clique. Put $F := F \cup \{C\}$ where $C = N_1[u]$. Perform Subroutine(C).

If some of the conditions in Subroutine and Routine does not hold, then either $G \notin L_3^l$ (if there exists a large clique in G), or we have to try another initialisation (if there is no large clique in G). If all initialisations fail, then $G \notin L_3^l$.

Theorem 12 . *Algorithm 11 is correct and can be implemented to run in time $\mathcal{O}(nm)$.*

Proof: The correctness follows by the results of the previous section.

We considered a graph to be defined by List of Adjacency in which the neighborhood of each vertex is ordered by the increment of indices of its vertices. But simultaniously we operate with Adjacency Matrix. We delete edges (i.e. change the current graph) simultaniously in List and Matrix. Matrix is used in Algorithm only for checking whether a is adjacent to b. In Matrix it requires a constant time whereas in List it can be done in $min\{deg(a), deg(b)\}$.

Finding a vertex of maximum degree requires linear time. Checking whether a 20-vertex graph contains some 8-clique requires only constant time. Extending a clique to a maximal clique can be done in time $\mathcal{O}(m)$. So Steps (1)–(3) require $\mathcal{O}(m)$ time.

We can find a leading vertex (Step (4.1)) in time $\mathcal{O}(n)$. Finding a good vertex in the neighborhood of a leading vertex (Step (4.2)) requires $\mathcal{O}(n)$ time. (Since checking whether a 0-weighted vertex is good requires only a constant time.) For a good vertex b, finding a b-good clique C_b (Step (4.3)) can be done in time $\mathcal{O}(m)$. Finding an (F, a)-pair (Step (4.4)) requires a constant time. Finding an (F, a)-tuple (Step (4.5)) can be done in time $\mathcal{O}(m)$ since in this case $N_2[a]$ contains a constant number of vertices. Adding a constant number of cliques to F requires a constant time.

After choosing a leading vertex, we either label it or increase the number of vertices covered by F. Each vertex is labelled at most once. Hence we perform Step (4.1) at most $2n$ times. So the operations performed by Routine properly (without Subroutine) require $\mathcal{O}(nm)$ time.

Now turn to Subroutine. Clearly, each of (S.1) and (S.5) can be implemented in time $\mathcal{O}(m)$ while (S.2)-(S.4) require $\mathcal{O}(n)$ time. Hence a single performing Subroutine can be done in time $\mathcal{O}(m)$. Step (S.3) guarantees that F contains at most $3n$ cliques. Subroutine is performed only if at least one clique is added to F. So during the algorithm Subroutine performs at most $3n$ times. Therefore the summarized time for performing Subroutine as many times as the algorithm prescribes is $\mathcal{O}(nm)$.

Line graphs can be recognized in linear time ([8], [16]).

Since the algorithm at worst has $\binom{18}{6}$ initialisations, it can be implemented in time $\mathcal{O}(nm)$. □

References

[1] Beineke L.W. Derived graphs and digraphs, *Beiträge zur Graphentheorie*, Teubner, Leipzig (1968) 17–33.

[2] Berge C. *Hypergraphs. Combinatorics of Finite Sets*, Amsterdam, 1989.

[3] Bermond J.C., Meyer J.C. Graphs représentatifs des arêtes d'un multigraphe, *J. Math. Pures et Appl.* 52 (1973) 299–308.

[4] Degiorgi D.G., Simon K. A dynamic algorithm for line graph recognition, WG95, LNCS1017.

[5] Hliněný P., Kratochvíl J. Computational complexity of the Krausz dimension of graphs, WG97, LNCS1335, 214–228.

[6] Jacobson M.S., Kezdy A.E., Lehel J. Recognizing intersection graphs of linear uniform hypergraphs, *Graphs and Comb.* (1997)

[7] Krausz J. Démonstration nouvelle d'un théoreme de Whitney sur les réseaux, *Mat.Fiz. Lapok* 50 (1943) 75–85.

[8] Lehot P.G.H. An optimal algorithm to detect a line graph and output its root graph, *J. Assoc. Comput. Mach.* 21 569–575.

[9] Levin A., Tyshkevich R. Line hypergraphs, *Discrete Math. Appl.* 3 (1993) 407–427.

[10] Lovász L. Problem 9, *Beiträge zur Graphentheorie und deren Anwendungen : Vorgetragen auf dem international kolloquium*, Oberhof (1977) P. 313.

[11] Metelsky Yu., Tyshkevich R. Line graphs of linear 3-uniform hypergraphs, *J. Graph Theory* 25 (1997) 243–251.

[12] Naik R.N., Rao S.B., Shrikhande S.S., Singhi N.M. Intersection graphs of k-uniform linear hypergraphs, *Ann. Discrete Math.* 6 (1980) 275–279.

[13] Naik R.N., Rao S.B., Shrikhande S.S., Singhi N.M. Intersection graphs of k-uniform linear hypergraphs, *Europ. J. Combin.* 3 (1982) 159–172.

[14] Naor J., Novick M.B. An efficient reconstruction of a graph from its line graph in parallel, *J. Algorithms* 11 (1990) 132–143.

[15] S. Poljak, V. Rödl, D. Turzik, Complexity of representation of graphs by set systems, Discrete Appl. Math. 3 (1981) 301-312.

[16] Roussopoulos N.D. A max {m,n} algorithm for determining the graph H from its line graph G, *Inform. Process. Lett.* 2 (1973) 108–112.