COMBINATORICA 3 (2) (1983) 141—146

HYPERGRAPHS WITH NO SPECIAL CYCLES

R. P. ANSTEE*

Received 16 September 1981
Revised 14 February 1983

A special cycle in a hypergraph is a cycle $x_1E_1x_2E_2x_3\dots x_nE_nx_1$ of n distinct vertices x_i and n distinct edges E_j $(n\geqq 3)$ where $E_i\cap\{x_1, x_2, \dots, x_n\}=\{x_i, x_{i+1}\}$ $(x_{n+1}=x_1)$. In the equivalent (0, 1)-matrix formulation, a special cycle corresponds to a square submatrix which is the incidence matrix of a cycle of size at least 3. Hypergraphs with no special cycles have been called totally balanced by Lovász. Simple hypergraphs with no special cycles on m vertices can be shown to have at most $\binom{m}{2}+m+1$ edges where the empty edge is allowed. Such hypergraphs with the maximum number of edges have a fascinating structure and are called solutions. The main result of this paper is an algorithm that shows that a simple hypergraph on at most m vertices with no special cycles can be completed (by adding edges) to a solution.

1. Introduction

The problem was originally inspired by a result of Ryser on (0, 1)-matrices without triangles [7]. We use hypergraph terminology throughout the paper in the hope that the arguments will be easier to visualize. Of course a (0, 1)-matrix A is equivalent to a hypergraph H where the rows index vertices and the columns index edges (sets of vertices). See Berge [5] for an introduction to hypergraphs. We only use elementary properties of hypergraphs and allow the empty edge $\emptyset$ which was forbidden by Berge.

We define a *special cycle* of size n $(n\geqq 3)$ to be a cycle $x_1E_1x_2E_2x_3\dots x_nE_nx_1$ of distinct vertices $x_1, x_2, \dots, x_n$ and distinct edges $E_1, E_2, \dots, E_n$ where $E_i\cap\{x_1, x_2, \dots, x_n\}=\{x_i, x_{i+1}\}$ for $i=1, 2, \dots, n$. This is stronger than the usual definition of a cycle and might be thought of as an "elementary" cycle. Throughout the paper, we will assume that the indices are cyclic in a special cycle of length n, i.e. $x_{n+1}=x_1$, $E_{n+1}=E_1$. Hypergraphs without special cycles correspond to (0, 1)-matrices containing no configurations C_k, for $k\geqq 3$, in the notation of [1]. These hypergraphs are called totally balanced by Lovász [6].

A *simple* hypergraph is one in which all the edges are distinct. In [1], it was shown that a simple hypergraph on m vertices, with no special cycles of size 3, has

---

* Support provided by NSERC.
AMS subject classification (1980): 05 C 65

at most $\binom{m}{2}$ edges of size at least 2. If it has $\binom{m}{2}$ such edges, then it has no special cycles of any size. In this paper, we consider simple hypergraphs with no special cycles. We do not restrict the edge sizes and, in particular, the empty edge and all edges of one vertex could be added to the hypergraph, if not present, without creating special cycles. Thus a *maximum* (simple) *hypergraph with no special cycles* has $\binom{m}{2}+m+1$ edges. The following is our main result.

Theorem 1.1. *Let K be a simple hypergraph on at most m vertices with no special cycles. Then K can be completed to a maximum hypergraph with no special cycles by adding edges.*

An alternate statement of Theorem 1.1 is that *maximal* hypergraphs with no special cycles are *maximum* hypergraphs with no special cycles. Further remarkable properties of these maximum hypergraphs appear in [1, 2, 3] and they seem to be an excellent generalization of trees. A referee pointed out that the edge sets without special cycles of a hypergraph do not define the independent sets of a matroid (as for trees). Thus the completion algorithm of Section 3, which is used to prove Theorem 1.1, is necessarily more involved than the matroid greedy algorithm. The algorithm is, however, seen to be polynomial in complexity.

2. Preliminary lemmas and overlaying

Hypergraphs with no special cycles have been shown to have a number of properties.

Lemma 2.1. ([1]) *Let H be a simple hypergraph on m vertices with n edges of size l ($l\geqq 2$) and with no special cycles. Then $n\leqq m-l+1$.*

It is an easy summation to verify that then the number of columns of column sum l ($l\geqq 1$) is $m-l+1$ for a solution of size m. In order to understand the structure, it is useful to consider k-trees [4]. We define them inductively in hypergraph terms. A k-tree on $k+1$ vertices is just an edge of all the $k+1$ vertices. To obtain a k-tree on n vertices from one on $n-1$ vertices, add an edge of $k+1$ vertices one of which is the additional vertex and set of the remaining k vertices are a subset of some edge of the k-tree. Thus a 1-tree is the usual tree. We allow a 0-tree for convenience. As defined a k-tree for $k\geqq 2$ may have special cycles. For example, the 2-tree $E_1=\{1, 2, 3\}$, $E_2=\{1, 2, 4\}$, $E_3=\{1, 3, 5\}$, and $E_4=\{2, 3, 6\}$ contains a special cycle of size 3. However the following is true.

Lemma 2.2. ([1]) *Let H be a simple hypergraph on m vertices consisting of $m-l+1$ edges of size l. Assume H has no special cycles. Then H is a $(l-1)$-tree.*

The number of edges in a $(l-1)$-tree on m vertices is seen to be $m-l+1$.

We claim that the inductive construction of a $(k-1)$-tree T (of edges of size k) can be carried out starting with any subset S, of edges of T, which form a $(k-1)$-tree. To prove this, let the edges of T be indexed $E_1, E_2, \ldots, E_t$ according to the order in which they were introduced to form the $(k-1)$-tree T in its original inductive construction. Let E_q be the edge with the smallest index in S. Run through the inductive construction of T until E_q is added where E_q intersects some edge E_p in $k-1$ vertices and $p<q$. It is a simple matter to verify that $E_p \setminus E_q$ is not a vertex of the edges in S since otherwise there would be too many edges on the vertices of S. Now add E_p to S and repeat this process until $E_1 \in S$. The remaining edges of T can be added to S in the following way. Run through the inductive construction of T, omitting steps that add edges already present in S. This proves the claim.

Our algorithm in Section 3 relies heavily on the idea of *overlaying* a $(k-2)$-tree with a $(k-1)$-tree as given in the following algorithm. C is a set of vertices which grows as the algorithm proceeds. It need not be an edge of the hypergraph.

Overlaying Algorithm.

Step 1. A $(k-2)$-tree is given where $k \geqq 2$. Set E to be the union of the edges of the $(k-2)$-tree where we require $|E| \geqq k$.

Step 2. Select a pair of edges E^*, E^{**} of size $k-1$ from the $(k-2)$-tree with $|E^* \cap E^{**}| = k-2$. Set $E' = E^* \cup E^{**}$, the first edge of the $(k-1)$-tree and set $C = E'$.

Step 3. Perform Step 4 until $C = E$, then stop.

Step 4. Select a pair of edges E^*, E^{**} of size $k-1$ from the $(k-2)$-tree with $E^* \subseteq C$, $|E^* \cap E^{**}| = k-2$, $E^{**} \setminus E^* = \{v\} \notin C$. Add the edge $E' = E^* \cup E^{**}$ to the $(k-1)$-tree and add v to C.

We note that Step 2 corresponds to Step 4 with $C = E^*$. Our above claim ensures that there is an edge E^{**} as desired to extend the $(k-2)$-tree. Also, in Step 4, the edge E' intersects C in E^* which has been previously covered by an edge of size k in C. Thus E' extends the $(k-1)$-tree. Hence the algorithm always terminates. For $k=2$, this is the usual inductive construction of a tree.

In the algorithm of Section 3, we use this algorithm with the proviso that the edges E' will be chosen from edges already in existence, if possible. Thus we do not combine Step 2 with Step 4 by setting $C = E^*$ since this removes some of the freedom of choice in choosing E' in Step 2.

We define a hypergraph H to be *intersection closed* if for every pair of edges E_1, E_2 of H, the edge $E_1 \cap E_2$ is in H. For this we consider $\emptyset$ to be a possible edge, corresponding in the matrix interpretation to a column of 0's. This property is of interest because, in [2], it was shown that maximum hypergraphs with no special cycles are intersection closed.

Lemma 2.3. *Let K be a hypergraph with no special cycles and let E', E'' be two edges of K. Then adding the edge $E' \cap E''$ to K does not create any special cycles.*

Proof. Assume $E_1 = E' \cap E''$ creates in K a special cycle $x_1 E_1 x_2 E_2 x_3 \dots x_n E_n x_1$. We have $x_3, x_4, \dots, x_n \notin E_1$ and thus each x_i $(i = 3, 4, \dots, n)$ can belong to at most one of E' or E''. Assume some are not in E'; say $x_{k+1}, x_{k+2}, \dots, x_{l-1} \notin E'$ and yet $x_k, x_l \in E'$ (recall $x_1, x_2 \in E'$). Then $x_l E' x_k E_k x_{k+1} \dots x_{l-1} E_{l-1} x_l$ is a special cycle in K, contradicting our hypothesis. This proves the lemma. ∎

3. Main algorithm

We now present the completion algorithm in order to prove Theorem 1.1. It generalizes the algorithm given at the end of [1] which solved the case that the hypergraph K had no edges.

Completion Algorithm.

Step 1. Let a simple hypergraph K on at most m vertices, with no special cycles, be given. For any pair of edges E', E'' of K, add to K the edge $E' \cap E''$ if it is not already present. Repeat until K is intersection closed.

Step 2. Add to K the empty edge $\emptyset$, all edges of one vertex, and the edge of all m vertices, if not already present.

Step 3. Initialize H to be K. We will no longer alter K. For each edge E in K of size at least 3, where the edges are considered in order of increasing size, perform step 4. Then stop, H is the desired maximum hypergraph with no special cycles.

Step 4. For $k=2, 3, \ldots, |E|-1$, use the overlaying algorithm to place a $(k-1)$-tree over a $(k-2)$-tree, both on the vertices of E. As the $(k-1)$-tree is formed, add the edges to H. Edges of size k, already present in H, are to be used wherever possible in the overlaying algorithm in preference to creating new edges.

Theorem 3.1. *The algorithm performs the completion described in Theorem 1.1.*

Proof. We assert that the algorithm terminates since each step will. For Step 1 we appeal to Lemma 2.1. For Step 4, we appeal to the fact that the overlaying algorithm always terminates. Thus when Step 3 is performed on the edge consisting of all m vertices, then H will have a k-tree for $k=0, 1, \ldots, m-1$ and thus a simple count ensures that H has enough edges to be a maximum hypergraph with no special cycles. If we show that H has no special cycles, then we will be done.

Applying Lemma 2.3, it is clear that Steps 1 and 2 leave K intersection closed with no special cycles. When H is initialized as K, H satisfies the following three conditions. Assume we have selected E in Step 3.

(i) If H has an edge E' strictly contained in E, then E' covers a k-tree (of edges of H) on its vertices for $k=0, 1, \ldots, |E'|-1$.
(ii) H has no special cycles.
(iii) H is intersection closed.

At this point (i) is trivial and (ii) and (iii) hold in K. We will verify that (i), (ii), (iii) hold for H after an edge is added in Step 4 if they hold for H before. This gives an inductive proof of the theorem.

Let Step 4 be in progress with E the edge of K currently under consideration from Step 3. Assume that we are in Step 2 or Step 4 of the overlaying algorithm about to add an edge E_1. We may assume (ii) and (iii) holds. The special nature of (i) requires that it be verified each time a new E is chosen and thereafter it may be assumed inductively. Say E has just been chosen in Step 3. Consider an edge E' strictly contained in E. If $E' \in K$, then Step 4 has been performed on it and so the claim is true. If E' has been added in the course of the algorithm as $E' = E^* \cup E^{**}$ in the overlaying algorithm, then $|E^*|=|E^{**}|=|E'|-1$. Inductively E^* and E^{**} cover $|E'|-k$ edges of size k for $k=0, 1, \ldots, |E'|-1$. Using (ii), we have that $E^* \cap E^{**} \in H$ and we know $|E^* \cap E^{**}|=|E'|-2$. Then $E^* \cap E^{**}$ covers $|E'|-k-1$ edges of size k for $k=0, 1, \ldots, |E'|-2$. But then $E^* \cup E^{**}=E'$ covers $|E'|-k+1$ edges of size k for $k=0, 1, \ldots, |E'|$. Appealing to Lemma 2.2 and using (iii), we find that these $|E'|-k+1$ edges form a $k-1$ tree. This verifies (i).

Let the edge to be added, E_1, be $E^* \cup E^{**}$ in the overlaying algorithm. We need to verify (i), (ii) and (iii) hold after E_1 is added. We can verify that (i) holds by showing that E_1 covers a k-tree for $k=1, 2, \ldots, |E_1|-1$ using the above argument.

The difficult part is showing that (ii) holds. Assume the contrary, that E_1 creates a special cycle, say

$$x_1 E_1 x_2 E_2 x_3 \ldots x_n E_n x_1. \tag{3.1}$$

Let $E_1=E^* \cup E^{**}$ in the overlaying algorithm which produced it with $|E^*|= =|E^{**}|=k-1$, $|E^* \cap E^{**}|=k-2$. Let C be the current value of C in the overlaying algorithm or in the case it is undefined, set $C=E^*$. In (3.3), we have $x_1, x_2 \in E$. If $x_{k+1}, x_{k+2}, \ldots, x_{l-1} \notin E$ and $x_k, x_l \in E$, then

$$x_k E_k x_{k+1} \ldots x_{l-1} E_{l-1} x_l E x_k \tag{3.2}$$

is a special cycle in H, contradicting our hypothesis. Thus we may assume $x_1, x_2, \ldots, x_n \in E$. Since H is intersection closed, $E_i \cap E$ is an edge for $i=2, 3, \ldots, n$.

We may replace E_i by $E_i \cap E$ in (3.2) and still have a special cycle. Thus we will assume $E_i \subseteq E$ for $i=1, 2, \ldots, n$. We deduce that $E_i \neq E$ for $i=1, 2, \ldots, n$ since this would violate the definition of a special cycle.

Recall that $E_1=E^* \cup E^{**}$. If $x_1, x_2 \in E^*$, then

$$x_1 E^* x_2 \ldots x_n E_n x_1 \tag{3.3}$$

would be a special cycle in H, contradicting our hypothesis. The same holds for E^{**}, so we deduce that $x_1, x_2 \notin E^* \cap E^{**}$. Let $x_1 \in E^* \setminus E^{**}$ and $x_2 \in E^{**} \setminus E^*$. Thus x_2 is the vertex v of Step 4 of the overlaying algorithm which produces E_1, i.e. $x_2 \notin C$. (We have set $C=E^*$ in the case that Step 2 produced E_1). We will show $x_2 \in C$ to obtain our contradiction.

We are going to show first that $E^* \cap E^{**} \subseteq E_i$ for $i=2, 3, \ldots, n$. This possibility was overlooked in [1]. Let $x \in E^* \cap E^{**}$. Then we have a cycle (not necessarily special)

$$x_1 E^* x E^{**} x_2 E_2 x_3 \ldots x_n E_n x_1. \tag{3.4}$$

If $x \notin E_2 \cap E_3 \cap \ldots \cap E_n$ then we can find a special cycle. Simply choose a maximal set of consecutive edges which do not contain x. Extend this set by the two edges on either side in the cycle and join them by x. This contradiction to (ii) proves that $x \in E_2 \cap E_3 \cap \ldots \cap E_n$ and so $E^* \cap E^{**} \subseteq E_i$ for $i=2, 3, \ldots, n$ as claimed.

We claim that if $\tilde{E} \subseteq C$ for any edge $\tilde{E}$ of size $k-1$ with $\tilde{E} \subseteq E_i$ then $E_i \subseteq C$, where i is between 2 and n. We have that E_i covers a $(k-1)$-tree and a $(k-2)$-tree on its vertices. First we are going to show that the overlaying algorithm can be performed starting with an arbitrary edge E' of the $(k-1)$-tree such that only edges of the $(k-1)$-tree are added. Now E' is an edge of size k and so covers two edges of size $k-1$. Thus E' can be used in Step 2 of the overlaying algorithm to start the $(k-1)$-tree. Consider some further stage where we have already selected a $(k-1)$-tree on the vertices of C. We know that we can extend the $(k-1)$-tree to the full $(k-1)$-

tree on the vertices of E_i, thus we may select an edge E'' of the $(k-1)$-tree on C and an edge E''' to extend the tree. Hence $|E''\cap E'''|=k-1$ and $E'''\setminus C=\{v\}$. Now H is intersection closed so $E''\cap E'''=\bar{E}$ is an edge of the $(k-2)$-tree. Also E''' covers two edges of the $(k-2)$-tree and so there is an edge $\bar{\bar{E}}$ of size $k-1$ with $\bar{E}\cup\bar{\bar{E}}=E'''$ and $\bar{\bar{E}}\setminus C=\{v\}$. Thus E''' can be chosen to extend the $(k-1)$-tree in Step 4 of the overlaying algorithm.

If we count up the edges of size $k-1$, we deduce that each such edge is covered by an edge of size k. Thus we may start the overlaying algorithm with any edge $\bar{E}$ of the $(k-2)$-tree as C by simply selecting an edge E' of the $(k-1)$-tree with $\bar{E}\subseteqq E'$ and using the above described construction. We may do the same construction for any $\bar{E}\subseteqq C$, omitting steps that add edges already present, to deduce that $E_i\subseteqq C$. This proves the claim.

In our case $E^*\subseteqq C$ and $E^*\subseteqq E_2$, so we deduce that $E_2\subseteqq C$. We have $x_3\in E_2\cap E_3$ and $E^*\cap E^{**}\subseteqq E_2\cap E_3$ where $x_3\notin E^*\cap E^{**}$ since (3.3) was a special cycle. Now H is intersection closed, so $E_2\cap E_3$ is an edge of size at least $k-1$ and so it covers an edge of size $k-1$. Thus, by our above arguments, $E_3\subseteqq C$. Similarly, $E_4\subseteqq C, \ldots, E_n\subseteqq C$. But then $x_2\in C$, our desired contradiction. Thus no special cycle, as in (3.3), was created and (ii) holds.

We must prove that after adding the edge E_1 to H, that (iii) holds. Consider $E'\cap E_1$. It is either E_1 or smaller than E_1. In the latter case, adding the edge $E'\cap E_1$ will not create any special cycles by Lemma 2.3. Yet E_1 covers a k-tree on its vertices for $k=0, 1, \ldots, |E_1|-1$, i.e. the maximum number by Lemma 2.1. Thus the edge $E'\cap E_1$ must already be present. We conclude that the resulting hypergraph is intersection closed.

Thus by induction, (i), (ii), and (iii) hold and thus at the termination of the algorithm, H has no special cycles. We have already shown that this proves the theorem. ∎

Acknowledgements. The author thanks the referees for comments which greatly improved the presentation of this paper.

References

[1] R. P. ANSTEE, Properties of (0, 1)-matrices with no triangles, *J. Comb. Theory (A)* **29** (1980), 186—198.
[2] R. P. ANSTEE, Properties of (0, 1)-matrices without certain configurations, *J. Comb. Theory (A)*, **31** (1981), 256—269.
[3] R. P. ANSTEE, Network flows approach for matrices with given row and column sums, Discrete Math. **44** (1983), 125—138.
[4] L. W. BEINEKE and R. E. PIPPERT, The number of labelled k-dimensional trees, *J. Comb. Theory* **6** (1969), 200—205.
[5] C. BERGE, *Graphs and Hypergraphs*, North-Holland, Amsterdam, 1973.
[6] L. LOVÁSZ, *Combinatorial Problems and Exercises*, North-Holland, Amsterdam (1979).
[7] H. J. RYSER, A fundamental matrix equation for finite sets, *Proc. Amer. Math. Soc.* **34** (1972), 332—336.

Richard Anstee

Department of Combinatorics and Optimization
University of Waterloo, Waterloo, N2L 3G1
Canada