Leave Graphs of Small Maximal Partial Triple Systems

Charles J. Colbourn

Department of Computer Science,
University of Waterloo,
Waterloo, Ontario, N2L 3G1
CANADA

Rudolf A. Mathon

Department of Computer Science,
University of Toronto,
Toronto, Ontario, M5S 1A1
CANADA

ABSTRACT

All graphs meeting the basic necessary conditions to be the leave graph of a maximal partial triple system with at most thirteen elements are generated. A hill-climbing algorithm is developed to determine which of these candidates are in fact leave graphs. Improved necessary conditions for a graph to be a leave graph are developed.

1. Candidates, leaves, and pseudoleaves

A ***triple system*** of order v and index λ, $B[3,\lambda;v]$, is a pair (V,B). V is a v-set of *elements* and B is a collection of 3-element subsets of V called *triples* or *blocks*. Every 2-element subset of V is contained in precisely λ blocks. When this latter condition is relaxed to require that every 2-subset appears in at most λ blocks, the result is a *partial* triple system $PB[3,\lambda;v]$. Finally, when no block can be added to a partial triple system without violating the requirements, it is termed *maximal.*

Maximal partial triple systems (mpts) have been widely studied, in part to determine the smallest order of (full) triple system into which a given mpts can be embedded (see, for example, [2]). Only the collection of pairs left uncovered by the mpts affects the embedding; hence we define the *leave* of an mpts to the a multigraph whose vertices are the elements of the mpts; two vertices are connected by $\lambda-s$ edges in the multigraph exactly when the corresponding pair of elements appear in precisely s blocks of the mpts. Two mpts's with the same leave have the same embedding. It is therefore of significant interest to attempt to characterize multigraphs which are leaves. However, even for $\lambda=1$, this problem is far from settled.

Some numerical necessary conditions are immediate. We view a B[3,λ;v] as an edge-partition of the complete multigraph λK_v into triangles. A PB[3,λ;v] with leave L is then a partition of λK_v-L into triangles. It follows that the multigraph λK_v-L has all vertex degrees even, and has 0 (mod 3) edges. The vertex degrees and number of edges in λK_v are easily determined; from these, we obtain two necessary conditions on L:

(1) L has all vertex degrees even iff v is odd or λ is even; L has all vertex degrees odd if v is even and λ is odd.

(2) L has 0 (mod 3) edges if $v\equiv 0,1$ (mod 3) or $\lambda\equiv 0$ (mod 3); L has 1 (mod 3) edges if $v\equiv 2$ (mod 3) and $\lambda\equiv 1$ (mod 3); L has 2 (mod 3) edges if $v\equiv 2$ (mod 3) and $\lambda\equiv 2$ (mod 3).

In addition, for the leave L of a maximal PB[3,λ;v], we have

(3) L is triangle-free.

Any multigraph on v vertices which is a subgraph of λK_v and satisfies conditions (1)-(3) for a specific λ is termed a λ*—candidate*, or simply *candidate* when $\lambda=1$. It is well known that not all candidates are leaves; such a graph is called a *pseudoleave* by Stinson and Wallis [10]. $C_4 \cup C_5$ is a small example of a pseudoleave. Colbourn and Rosa [5] constructed an infinite class of pseudoleaves, and Stinson and Wallis later established that pseudoleaves exist for most of the admissible numbers of edges [10].

In this paper, we generate all nonisomorphic candidates on at most thirteen vertices, and determine which are leaves and which are pseudoleaves. In the process, we develop a powerful hill-climbing technique for determining whether a candidate is a leave. We also improve the necessary conditions for a candidate to be a leave.

2. Generating the candidates

One cannot hope to consider all nonisomorphic graphs on at most thirteen vertices to determine which are candidates. The largest order for which graphs have been exhaustively generated is ten [1], and there are already 12,005,168 nonisomorphic graphs here. Hence we generated the candidates directly. To do this, we observe that only a small fraction of the possible degree sequences for graphs are degree sequences for candidates, in view of requirement (1). Generating integer sequences meeting requirements (1) and (2) is straightforward; we then used the Havel-Hakimi algorithm [7,8] to eliminate those with no graphical realization. All sequences which remain have some graphical realization meeting requirements (1) and (2), but not necessarily requirement (3).

The next step is to generate graphs with specified degree sequence, and which contain no triangles. To do this, we use the modification of Farrell's algorithm [6] due to Colbourn and Read [4]. As a graph is generated, a canonicity check is made to ensure that the graph is nonisomorphic to all graphs generated thus far; this canonicity check is implemented using an algorithm due to Mathon. This generation algorithm generates precisely the nonisomorphic candidates.

The method employed here could be improved somewhat if one had a useful characterization of degree sequences which have triangle-free realizations. Further improvements could also be made by incorporating isomorph rejection at earlier steps in Farrell's algorithm. Nevertheless, using the algorithm outlined here, we generated all candidates on at most 13 vertices. The last case, v=13, consisted of 3234 candidates.

3. A hill-climbing algorithm

Stinson [9] developed a clever hill-climbing method for constructing random Steiner triple systems. The basic hill-climbing step in Stinson's algorithm operates as follows. Suppose that B is a partial triple system on the element set V; if (V,B) has nonempty leave, there is (at least) one pair of incident edges in the leave. Let us suppose that $\{x,y\}$ and $\{x,z\}$ are in the leave. If $\{y,z\}$ is also an edge of the leave, we simply add the triple $\{x,y,z\}$ to B, thus forming a partial triple system with one more triple. Otherwise, we locate the triple $\{w,y,z\}$ containing $\{y,z\}$ and replace it with $\{x,y,z\}$, producing a partial triple system with the same number of triples as the original. While it is possible in principle for this method to get "stuck" by producing a nonextendible partial triple system, this seems never to happen in practice.

We modify Stinson's algorithm as follows, in order to test whether a candidate graph appears as a leave. We mark all edges in the candidate as *required*, and never allow a triple to use one of these edges. As in Stinson's method, we select a pair $\{x,y\}$, $\{x,z\}$ of incident edges. There are three possibilities for the edge $\{y,z\}$. If it is unused and not required, we add the triple $\{x,y,z\}$. If it is used by a triple $\{w,y,z\}$, we replace this triple by $\{x,y,z\}$. The third possibility is that it is required; in this event, we abandon this choice of incident edges and randomly select another pair. We require another modification to Stinson's method as well. In his method, one need not be concerned that the desired object does not exist; however, once certain edges are marked as required for the leave, there might be no maximal partial triple system with this candidate leave. Hence we stipulate an upper bound on the number of trials for pairs of incident edges at each level before progress is made by adding a triple; if this upper bound is exceeded, the hill-climbing is abandoned. This modified method does get "stuck", even when there is a solution. However, we found it better to do a small number (say, twenty) of independent hill-climbs rather than to increase the upper bound on time spent at a level. In our examples, most leaves were detected by the first hill-climb, and virtually all were detected within the first two.

It is perhaps remarkable that the hill-climbing strategy is so effective even with the number of added constraints. We believe that it is an indication of the power of randomizing a computation, in an effort to avoid getting mired down in an unprofitable subcase. Undoubtedly, the success is also an indication of the large number of partial triple systems that correspond to a given leave.

4. Small Orders

We summarize here the results of a complete enumeration of candidates on at most thirteen vertices. For each number of vertices, and each admissible number of edges, we give the number of nonisomorphic graphs which are candidates, and the number which are leaves and pseudoleaves.

Candidates of order 7				
# edges	0	6	9	total
Candidates	1	1	1	3
Leaves	1	1	0	2
Pseudoleaves	0	0	1	1

Candidates of order 8				
# edges	4	7	10	total
Candidates	1	3	2	6
Leaves	1	3	0	4
Pseudoleaves	0	0	2	2

Candidates of order 9					
# edges	0	6	9	12	total
Candidates	1	1	4	7	13
Leaves	1	1	3	4	9
Pseudoleaves	0	0	1	3	4

Candidates of order 10							
# edges	6	9	12	15	18	21	total
Candidates	1	8	9	14	2	2	36
Leaves	1	8	7	5	0	1	22
Pseudoleaves	0	0	2	9	2	1	14

Candidates of order 11									
# edges	4	7	10	13	16	19	22	25	total
Candidates	1	1	8	30	51	16	5	2	114
Leaves	1	1	8	28	38	7	2	0	85
Pseudoleaves	0	0	0	2	13	9	3	2	29

Candidates of order 12										
# edges	6	9	12	15	18	21	24	27	30	total
Candidates	1	6	21	97	213	135	29	13	1	516
Leaves	1	6	21	97	198	91	13	9	0	436
Pseudoleaves	0	0	0	0	15	44	16	4	1	80

Candidates of order 13													
# edges	0	6	9	12	15	18	21	24	27	30	33	36	total
Candidates	1	1	4	27	137	802	1175	865	179	32	7	3	3234
Leaves	1	1	4	27	137	799	1150	778	109	16	2	3	3028
Pseudoleaves	0	0	0	0	0	3	25	87	70	16	5	0	206

A complete set of diagrams of all pseudoleaves on at most twelve vertices is given in the appendix.

5. Necessary Conditions

Two further elementary necessary conditions have been used previously to establish that a candidate is not a leave. The first, a density condition, was used in [3,5]:

Lemma 1: Let G be a candidate on v vertices and e edges, having an edge-cutset of c edges which separates G into components of sizes s and $v-s$. If G is a leave, then

$$\binom{v-s}{2} + \binom{s}{2} - e + c \geq \tfrac{1}{2}(s(v-s) - c).$$

Lemma 1 rules out candidates such as $C_4 \cup C_5$, and numerous others. A different approach was taken by Stinson and Wallis [10], who observed that for G to be a leave, every edge in $\overline{G}$ must appear in a triangle; moreover, if an edge appears in a unique triangle, this requirement holds again for $\overline{G}$ with that triangle removed.

It is our purpose here to unify (and strengthen) these two approaches. To do this, we consider conditions on the graph $\overline{G}$ which are necessary for $\overline{G}$ to have an edge-partition into triangles, that is, for G to be a leave. We examine the *neighbourhood* $N(X)$ of a set X of vertices, that is, the set of all vertices not in X which are adjacent to one or more vertices of X. We call a pair of sets (X,Y) a *fence* for the graph $\overline{G}=(V,E)$ if $X,Y \subseteq V$, $X \cap Y = \emptyset$, and $\{N(X)\backslash Y\} \cap \{N(Y)\backslash X\} = \emptyset$. The *fence-degree* $fd(x)$ of a vertex $x \in X \cup Y$ is $|N(x) \cap (X \cup Y)|$. The *X-defect* $def_X(X,Y)$ of the fence (X,Y) is the minimum number of edges in a partial subgraph H of the subgraph induced on X which has $deg_H(x) \equiv fd(x) \pmod 2$ $\forall x \in X$ (a matching on the vertices of odd fence-degree is minimum if one exists); the *Y-defect* $def_Y(X,Y)$ is defined similarly. The *defect* $def(X,Y)$ is then the sum of the X-defect and the Y-defect.

Finally, for a fence (X,Y), let $\epsilon(X,Y)$ equal the number of edges connecting vertices of X to vertices of Y, and let $\iota(Z)$ be the number of edges connecting vertices within some set $Z \subseteq V$. We establish a relation between the number of "crossing" edges, the number of "inside" edges, and the defect.

Theorem 2: Let (X,Y) be a fence of $\overline{G}$. G is a leave only if $\epsilon(X,Y)$ is even, and

$$\epsilon(X,Y) \geq 2(\iota(X) + \iota(Y) - def(X,Y)).$$

Proof:

Consider those triangles which contain the edges having one endpoint in X and the other in Y. Since (X,Y) is a fence, the third node must lie either in X or in Y. Thus immediately $\epsilon(X,Y) \equiv 0 \pmod 2$ is required. For every two edges counted by $\epsilon(X,Y)$, one edge inside Y or inside X is accounted for. Removing all of these triangles leaves the parity of all fence-degrees unchanged modulo 2. Hence the number of edges inside X which must remain is (at least) $def_X(X,Y)$; similarly for Y. □

We should remark that Lemma 1 is an easy consequence of this theorem. In addition, edges belonging to zero triangles are ruled out; for such an edge, say $\{x,y\}$, taking $X = \{x\}$ and $Y = \{y\}$ in the theorem yields a violated inequality. The power of these inequalities in eliminating candidates is somewhat difficult to test, but these extreme examples show the generality of the theorem.

6. Concluding Remarks

A number of interesting questions arise from the research here. First, the hill-climbing method developed suggests that partitioning a graph into triangles is, in practice, a relatively easy task; however, theoretical results suggest that it is hard. It would be interesting to establish that the algorithm described succeeds on random graphs with high probability.

Second, the computational results suggest that the ratio of pseudoleaves to candidates approaches zero as n increases; a proof of this would be quite interesting. Third, the necessary condition developed in section 5 ensures that certain density requirements must be met. It does not appear likely that the condition given there can be tested in polynomial time, however. Hence, there is reason to suspect that conditions of this form may in fact be sufficient conditions (although the current conditions are too weak for this).

Acknowledgements

This research was undertaken while the first author was visiting the Department of Computer Science, University of Auckland, New Zealand, and the second author was visiting the Basser Department of Computer Science, University of Sydney, Australia. Special thanks to Peter Gibbons for all of his help.

References

[1] R.D. Cameron, C.J. Colbourn, R.C. Read, and N.C. Wormald, "Cataloguing the graphs on 10 vertices", *J. Graph Theory* 9 (1985) 551-562.

[2] C.J. Colbourn, R.C. Hamm, C.C. Lindner, and C.A. Rodger, "Embedding partial graph designs, block designs, and triple systems with $\lambda > 1$", *Can. Math. Bull.* 29 (1986) 385-391.

[3] C.J. Colbourn and B.D. McKay, "Cubic neighbourhoods in triple systems", *Ann. Discrete Math.*, to appear

[4] C.J. Colbourn and R.C. Read, "Orderly algorithms for generating restricted classes of graphs", *Int. J. Comput. Math.* A7 (1979) 167-172.

[5] C.J. Colbourn and A. Rosa, "Quadratic leaves of maximal partial triple systems", *Graphs and Combinatorics* 2 (1986) 317-337.

[6] E.J. Farrell, "Computer implementations of an algorithm to generate subcompositions and its application to a problem in graph theory", M.Math. thesis, University of Waterloo, 1971.

[7] S. Hakimi, "On the realizability of a set of integers as degrees of the vertices of a graph", *J. SIAM* 10 (1962) 496-506.

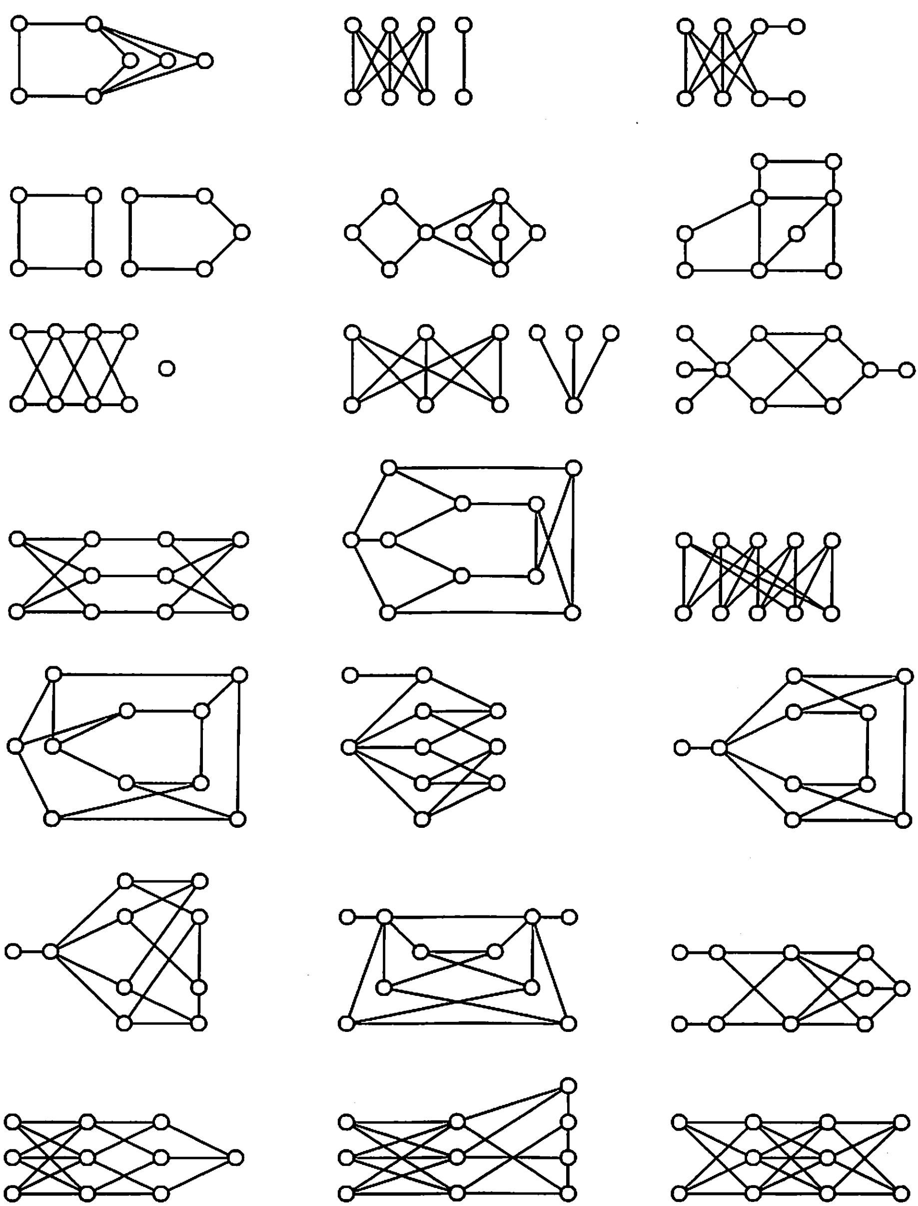
[8] V. Havel, "A remark on the existence of finite graphs" (in Hungarian), *Casopis Pest. Mat.* 80 (1955) 477-480.

[9] D.R. Stinson, "Hill-climbing algorithms for the construction of combinatorial designs", *Ann. Discrete Math.* 26 (1985) 321-334.

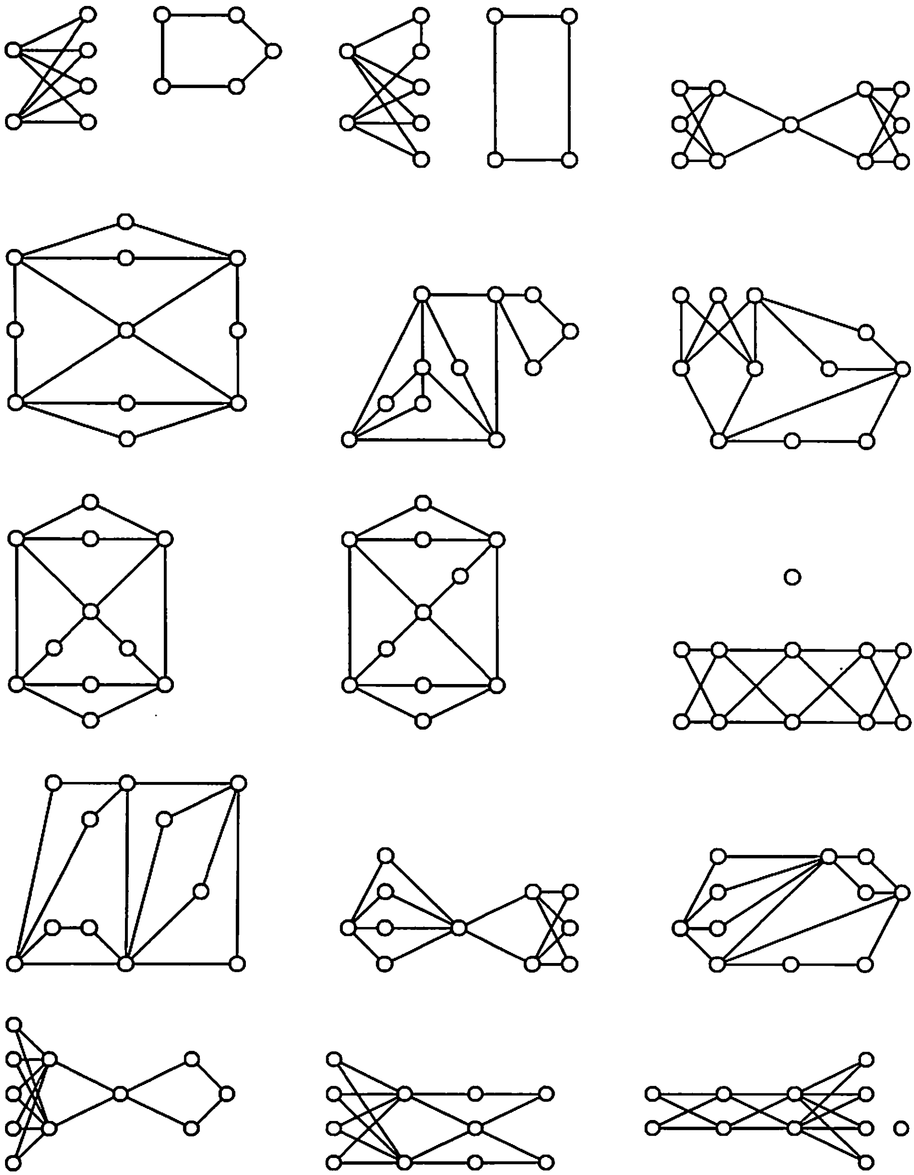
[10] D.R. Stinson and W.D. Wallis, "Graphs which are not leaves of maximal partial triple systems", *Ann. Discrete Math.*, to appear.

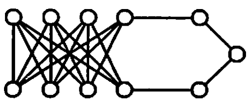
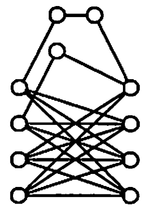
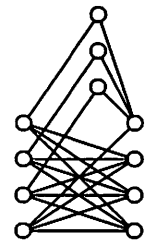
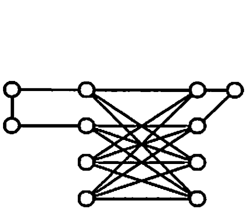
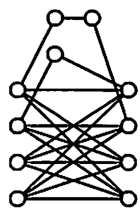
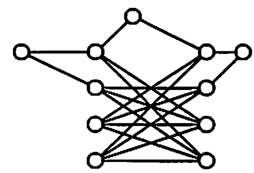
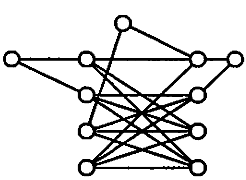
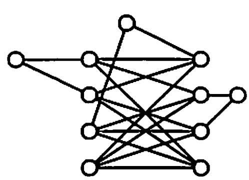
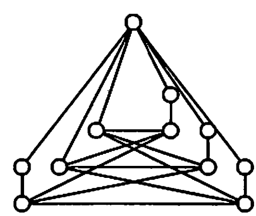
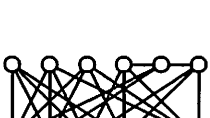
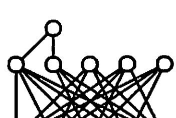
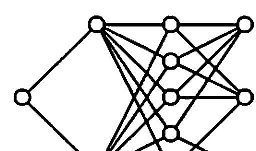
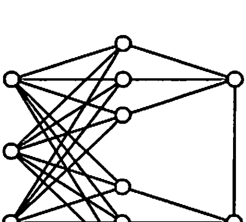
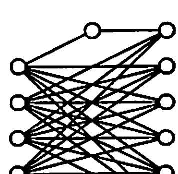
Appendix: Pseudoleaves on at most 12 vertices

$7 \leq v \leq 10$:



$v = 11$:



v = *12*: