

LSE Research Online

Frans J.C.T. de Ruiter, Norman L. Biggs Applications of integer programming methods to cages

Article (Published version) (Refereed)

Original citation:

de Ruiter, Frans and Biggs, Norman (2015) *Applications of integer programming methods to cages*. [The Electronic Journal of Combinatorics](#), 22 (4). P4.35. ISSN 1077-8926

© 2015 The Authors

This version available at: <http://eprints.lse.ac.uk/65785/>

Available in LSE Research Online: March 2016

LSE has developed LSE Research Online so that users may access research output of the School. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Users may download and/or print one copy of any article(s) in LSE Research Online to facilitate their private study or for non-commercial research. You may not engage in further distribution of the material or use it for any profit-making activities or any commercial gain. You may freely distribute the URL (<http://eprints.lse.ac.uk>) of the LSE Research Online website.

Applications of integer programming methods to cages

Frans de Ruiter*

Department of Econometrics and Operations Research
Tilburg University
Tilburg, The Netherlands

f.j.c.t.deruiter@tilburguniversity.edu

Norman Biggs

Department of Mathematics
London School of Economics and Political Science
London, U.K.

n.l.biggs@lse.ac.uk

Submitted: Nov 26, 2014; Accepted: Nov 12, 2015; Published: Nov 27, 2015
Mathematics Subject Classifications: 05C25, 05C35

Abstract

The aim of this paper is to construct new small regular graphs with girth 7 using integer programming techniques. Over the last two decades solvers for integer programs have become more and more powerful and have proven to be a useful aid for many hard combinatorial problems. Despite successes in many related fields, these optimisation tools have so far been absent in the quest for small regular graphs with a given girth. Here we illustrate the power of these solvers as an aid to construct small regular girth 7 graphs from girth 8 cages.

Keywords: Integer programming applications; cage problem

1 Introduction

The *girth* of a graph is the length of its shortest cycle. For the cage problem one aims to construct regular graphs of degree k with minimum number of vertices required to have girth g . It is known that a regular graph with degree k and girth g can be created for any $k \geq 2$ and $g \geq 3$ [3]. The real challenge is to find a regular graph with the least number

*Part of this author's work was done while he studied for the degree of M.Sc. in Applicable Mathematics at the Department of Mathematics, London School of Economics and Political Science.

of vertices which still has girth g . The graphs that are of minimum order for degree k and girth g are referred to as (k, g) -cages. Often, the constructions are not known to be cages and the resulting (k, g) -graphs only provide upper bounds, which we call *record graphs*. For an overview of the known cages, record graphs and their constructions see the up-to-date dynamic cage survey [5]. Only little appears to be known for the girth 7 case. In this paper we construct new record graphs with girth 7, improving the upper bounds for regular girth 7-graphs with degrees up to 14. This is achieved by a novel approach that uses the power of modern linear programming solvers and combinatoric properties of some known girth 8 graphs.

The only known cages with girth 7 are the McGee graph [11] of order 24 and the more recently discovered $(4, 7)$ -cage of order 67 [5, 6]. For all other degrees and girths only record graphs, which give an upper bound on the order of the cages, are known. This paper contributes to the girth 7 case of the cage problem by showing how to use integer programming techniques to construct record graphs with girth 7 for degrees up to 14. Table 1 lists the order of our new record graphs, as well as the previous best-known record graphs.¹

Table 1: Our new constructed graphs compared to old record graphs.

Degree k	New record	Old record	Old record due to
7	632	640	Exoo [4]
8	777	799	Sauer [14]
9	1104	1152	Araujo-Pardo [1]
10	1611	1639	Sauer [14]
11	2576	2596	Araujo-Pardo [1]
12	2893	2927	Sauer [14]
13	4292	4316	Araujo-Pardo [1]
14	4719	4759	Sauer [14]

2 The excision method

In this section we show how to construct regular graphs with girth $g - 1$ from regular graphs with girth g by *excision*. In an excision operation one first removes vertices and their adjacent edges from a graph with degree k and girth g . In the second step one has to join the remaining vertices with each other to obtain a regular graph of degree k and girth $g - 1$. We give results for the size of the set of vertices that can be removed to end up with a regular graph with girth $g - 1$. For joining up the vertices again, we use integer programming methods. For girth 7 there is a rather standard result that ensures us that

¹The adjacency lists of all these graphs, including the known girth 8 record graphs, are available online at <https://github.com/FransdR/Record-graphs-girth-7-and-8.git>

for odd k we can remove $2k$ vertices from a k -regular graph with girth 8. The proof for odd k and even girth g is given in [1]. Another proof can be based on the excision proof for the cubic situation [2]. Typical for those proofs is that the excised set of vertices and edges form a tree and the diameter of the tree is 3. An example of such a tree for degree 5 and girth 7 is given in Figure 1. Joining up the vertices to obtain a regular graph of

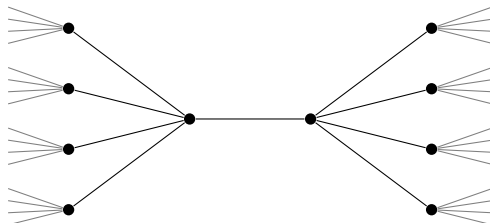


Figure 1: Subtree consisting of 10 vertices that can be removed from a $(5, 8)$ -graph to obtain a $(5, 7)$ -graph.

degree k can be done in an arbitrary way, as long as the joined vertices shared a common neighbour in the the excised tree. An example that can be obtained in this way is the McGee graph ($k = 3$) [11]. For the girth 7 case with higher degrees ($k \geq 4$) we might be able to excise a graph that consists of more than $2k$ vertices. Examples where this is successful are the cases $k = 5$ and $k = 6$ [12]. It is possible to excise 18 vertices in both cases. In the following sections we show how integer programming methods can be used to successfully excise larger sets of vertices for degrees 7 up to 14. We first describe the set of excised vertices and integer programming methods we developed for odd degrees 7, 9, 11 and 13 and after then a similar method for even degrees 8, 10, 12 and 14. Throughout this paper we denote V as the set of vertices in the original graph, S the set of vertices we are going to excise and $V \setminus S$ the vertices that remain after excision.

3 Odd degree graphs

For odd degree graphs we are going to excise a tree that has diameter 4, instead of a tree with diameter 3 as was done in the more general results [1]. Let us first explain why we can not join the edges from two neighbours in an arbitrary way if the diameter of S is 4. Consider two vertices u and v that are at distance 4 in S . Let $\{u_1, \dots, u_{k-1}\}$ and $\{v_1, \dots, v_{k-1}\}$ be the neighbours of respectively u and v that are still in $V - S$ and therefore have degree $k - 1$. There is path of length 6 going through S from any u_i to any other v_j . Hence, there could be a path through $V \setminus S$ of length 2 from u_i to v_j . Let $d(x, y)$ denote the distance between vertex x and y only using nodes from $V \setminus S$. If $d(u_1, v_1) + d(u_2, v_2) = 4$, then we are not allowed to add both edges $\{u_1, u_2\}$ and $\{v_1, v_2\}$ in the excision operation because that would create a 6-cycle. If this is the case, we call $(\{u_1, u_2\}, \{v_1, v_2\})$ a *forbidden combination*. Let \bar{E} be the pairs of vertices $\{x, y\}$ such that both x and y have degree $k - 1$. We have to choose edges from \bar{E} to create a regular graph of degree k . The set $R \subset \bar{E} \times \bar{E}$ that contains all forbidden combination of edges

is given by

$$R = \left\{ (\{x, y\}, \{w, z\}) \in \bar{E} \times \bar{E} \mid d(x, w) + d(y, z) = 4 \text{ or } d(x, z) + d(w, y) = 4 \right\}. \quad (1)$$

The set R only consists of pairs of edges, so we did not include any forbidden triples, quadruples etc. We do not have to mark any combinations of three or more edges as forbidden because every path through vertices from $V \setminus S$ joining u_i and v_j has length at least 2. Hence, a cycle with three or more ‘new’ edges from \bar{E} has length at least $2 + 2 + 2 + 1 + 1 + 1 = 9$.

We use the set of forbidden combination of edges R to determine whether we can join edges after excising the subtree S to obtain another regular graph of degree k . Excision of S is successful if we can determine an edge set $X \subset \bar{E}$ such that:

- (i) After adding edges from X , the degree of every vertex is k .
- (ii) For every edge pair $(e, f) \in R$, at most one of the edges e or f is in X .

To model these conditions we introduce decision variables x_e for all \bar{E} to decide whether or not to include an edge $e \in \bar{E}$:

$$x_e = \begin{cases} 1 & \text{if } e \in X \\ 0 & \text{if } e \notin X. \end{cases}$$

Using these decision variables, we can now translate the degree condition (i) into

$$\sum_{e \in \delta(u) \cap \bar{E}} x_e = 1 \quad \forall u \in N(S) \setminus S,$$

where $\delta(u)$ is the set of edges incident with vertex u and $N(S) \setminus S$ being the set of vertices that have a neighbour in S , but are not in S themselves. Note that the degree of every vertex in $N(S) \setminus S$ is equal to $k - 1$ before adding edges. So when every one of those vertices is adjacent to exactly one new edge from \bar{E} , condition (i) is satisfied. The second condition can be modelled as

$$x_e + x_f \leq 1 \quad \forall (e, f) \in R. \quad (2)$$

Hence, the resulting integer programming model is

$$\begin{aligned} & \max && 1 \\ & \text{such that} && \sum_{e \in \delta(u) \cap \bar{E}} x_e = 1 \quad \forall u \in N(S) \setminus S && \text{(Excision-ILP)} \\ & && x_e + x_f \leq 1 \quad \forall (e, f) \in R. \end{aligned}$$

We have used program (Excision-ILP) for odd degrees $3 \leq k \leq 13$. All solutions are obtained using the commercial solver Gurobi 5.6.2 [9] programmed in the YALMIP language [10] in Matlab on a 64-bit Mac OS X 10.9.4 PC with a 1.8 Ghz Intel Core i5 processor and 4GB of RAM. The initial girth 8 graphs were obtained using the labeling process for generalized quadrangles as described in [8] for graphs with degree $k = q + 1$, where q is a prime power, or the t -good structures from [7] for graphs with $k = q$. Starting from a $(k, 8)$ -graph where k is odd, we tried to excise different trees with diameter 4. For example, from the $(5, 8)$ -graph we successfully excised the following tree:

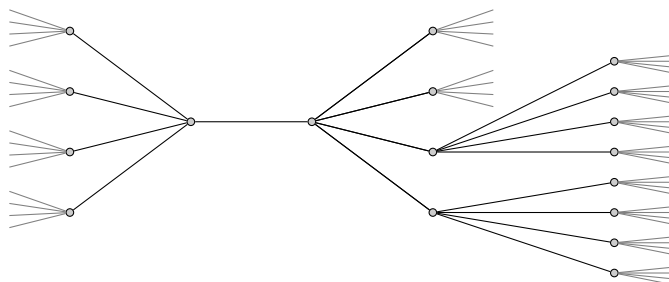


Figure 2: Tree S with diameter 4 consisting of 18 vertices that was successfully excised from the $(5, 8)$ -cage.

We have tried to include more vertices, but then the program did not return a feasible solution. This was either because it determined that the program was infeasible, or because we manually aborted it after a certain period of time. The only situations where we had to manually abort the program was for some sets S for degrees 11 and 13, where we stopped the solver after 60 minutes. This is because in all other cases, the solver returned a solution within 16 minutes. The results from our computations are displayed in Table 2.

Table 2: Results for individual excision on $(k, 8)$ -graphs for odd k .

Degree k	Order of $(k, 8)$ -graph	Order of $(k, 7)$ -graph	$ S $	$ R $	Time needed by Gurobi (sec)	Total time
3	30	24	6	0	< 1	< 1
5	170	152	18	198	< 1	2
7	658	632	26	899	4	9
9	1170	1104	66	37632	58	79
11	2618	2576	42	9267	107	228
13	4342	4292	50	20743	915	1391

Looking at the total running time, Gurobi clearly has the largest share. The total time depicted further includes: deleting the vertices and edges, calculating distance matrices, determine the forbidden combinations R and setting up the model in YALMIP. Especially calculating the distance matrices might take considerably more time for graphs with more vertices. Note that for degree 9 we could excise a bigger subset of vertices than for degree

11 and 13. This is probably due to the fact that the $(9, 8)$ -graph has a particularly nice structure as the incidence graph of a generalized quadrangle.

4 Even degree graphs

For even degrees, we can do a similar kind of excision to the method for odd degrees as done in Section 3. However, the excised set of vertices can not be a tree as we did in the odd graphs (see Figure 2) because we can only join an even number of neighbours by edges. When degree k is even, one would need to join $k - 1$ neighbours when excising a tree, which is an odd number. For even degrees $6 \leq k \leq 14$ we therefore excise a different set of size $3k - 1$. Only for the smallest even degree $k = 4$ we could not excise more than $3k - 2 = 10$ vertices. Note that for that degree the actual cage is known, which is not obtained via excision and contains 13 vertices less than the $(4, 8)$ -cage. For the even degrees $6 \leq k \leq 14$ the set of vertices that we remove is constructed as follows:

- Select a pair of vertices $\{u, v\}$ at distance 4 from each other that have exactly k vertices in common at distance 2.
- Remove u, v , their neighbours $N(u)$ and $N(v)$ and all but three of the vertices in $N_2(u, v)$, where $N_2(u, v)$ is the set of vertices that are at distance 2 from both u and v .

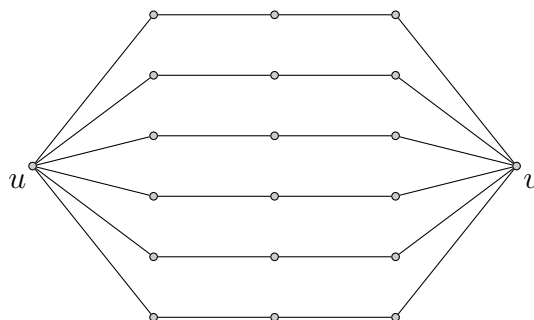


Figure 3: The set of excised vertices S from the $(6, 8)$ -graph consists of vertices u and v , their neighbours $N(u)$ and $N(v)$ and all but three of the vertices that are at distance 2 of both u and v .

An example of such an excised set for degree $k = 6$ is given in Figure 3. Note that for general even degree, we have no guarantee that there exist vertices u, v with $|N_2(u, v)| = k$. However, in case $k = q + 1$ and q is a prime power, we are guaranteed that every vertex pair $\{u, v\}$ at distance 4 has $|N_2(u, v)| = k$. This is due to the fact that every so-called point-pair in a generalized quadrangle is regular. For a more thorough description on point-pairs we refer the reader to the book by Payne and Thas [13].

The next step is joining up the vertices so that every vertex has degree k again. Let

S be the set of vertices consisting of $N(v)$, $N(v)$ and $N_2(u, v)$. Then every vertex in S has $k - 2$ neighbours in $V \setminus S$. Since k is an even number, $k - 2$ is even as well. The vertices in $N_2(u, v)$ that are not removed also have degree $k - 2$ since they have one edge adjacent to $N(u)$ and one to $N(v)$. For all degrees $k \geq 6$ we have to leave exactly 3 vertices from $N_2(u, v)$ in the remaining graph.

The constraints we add and the set of forbidden combinations $R \subset \bar{E} \times \bar{E}$ are similar to the ones in program (Excision-ILP). The paths through $V \setminus S$ that join two vertices in S are of length at least 2. Hence, every path containing three or more edges from R has length 9 again. Therefore, we can use the same definition for R , the set containing these forbidden combinations, as in (1). The only significant difference is that there are now vertices with degree $k - 2$, which need to be assigned two new edges instead of one. The computational results are given in Table 3, in the same format as for odd degrees. In this table we have again $|S|$ as the size of the removed set of vertices and $|R|$ number of forbidden combinations. For all but the degrees 4 and 6 our results give new record graphs that improve the upper bound on the order of the cages previously known in the literature.

Table 3: Results for individual excision on $(k, 8)$ -graphs for even k .

Degree k	Order of $(k, 8)$ -graph	Order of $(k, 7)$ -graph	$ S $	$ R $	Time needed by Gurobi (sec)	Total time
4	80	70	10	12	< 1	< 1
6	312	295	17	288	< 1	3
8	800	777	23	989	8	15
10	1640	1611	29	3108	37	68
12	2928	2893	35	7555	164	301
14	4760	4719	41	15625	1004	1554

References

- [1] G. Araujo-Pardo. On upper bounds of odd girth cages. *Discrete Mathematics*, 310(10-11):1622 – 1626, 2010.
- [2] N. L. Biggs. Constructions for cubic graphs with large girth. *Journal of Combinatorics*, 5:1–26, 1998.
- [3] Paul Erdős and Horst Sachs. Reguläre graphen gegebener taillenweite mit minimaler knotenzahl. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, 12:251–257, 1963.
- [4] G. Exoo. personal communication, July 2014.
- [5] Geoffrey Exoo and Robert Jajcay. Dynamic cage survey. *The Electronic Journal of Combinatorics*, 16, July 2013.

- [6] Geoffrey Exoo, Brendan D McKay, Wendy Myrvold, and Jacqueline Nadon. Computational determination of $(3, 11)$ and $(4, 7)$ cages. *Journal of Discrete Algorithms*, 9(2):166–169, 2011.
- [7] András Gács and Tamás Héger. On geometric constructions of (k, g) -graphs. *Contributions to Discrete Mathematics*, 3(1), 2008.
- [8] C. Godsil and G.F. Royle. *Algebraic Graph Theory*. Graduate Texts in Mathematics. Springer New York, New York, NY, 2001.
- [9] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2013.
- [10] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [11] WF McGee. A minimal cubic graph of girth seven. *Canadian Mathematical Bulletin*, 3:149–152, 1960.
- [12] Brendan McKay and Yang Yuanshen. Electronic adjacency lists, accessed online at <http://staffhome.ecm.uwa.edu.au/~00013890/remote/cages/allcages.html> on 3 July 2014.
- [13] Stanley E Payne and Joseph Adolf Thas. *Finite Generalized Quadrangles*, volume 110. European Mathematical Society Publishing House, Zurich, 2 edition, 2009.
- [14] N Sauer. Extremaleigenschaften regulärer graphen gegebener taillenweite, i and ii. *Sitzungsberichte Österreich. Akad. Wiss. Math. Natur. Kl.*, 176:9–25; 176 (1967), 27–43, 1967.