Constraints (2019) 24: 1–24
https://doi.org/10.1007/s10601-018-9294-5

Constraints for symmetry breaking in graph representation

Michael Codish[1] · Alice Miller[2] · Patrick Prosser[2] · Peter J. Stuckey[3]

Published online: 25 August 2018

Abstract

Many complex combinatorial problems arising from a range of scientific applications (such as computer networks, mathematical chemistry and bioinformatics) involve searching for an undirected graph satisfying a given property. Since for any possible solution there can be a large number of isomorphic representations, these problems can quickly become intractable. One way to mitigate this problem is to eliminate as many isomorphic copies as possible by *breaking* symmetry during search - i.e. by introducing constraints that ensure that at least one representative graph is generated for each equivalence class, but not the entire class. The goal is to generate as few members of each class as possible - ideally exactly one: the symmetry break is said to be *complete* in this case. In this paper we introduce novel, effective and compact, symmetry breaking constraints for undirected graph search. While incomplete, these prove highly beneficial in pruning the search for a graph. We illustrate the application of symmetry breaking in graph representation to resolve several open instances in extremal graph theory. We also illustrate the application of our approach to graph edge coloring problems which exhibit additional symmetries due to the fact that the colors of the edges in any solution can be permuted.

Keywords Constraints and satisfiability · Symmetry and search · Graph theory

✉ Alice Miller
alice.miller@gla.ac.uk

Michael Codish
mcodish@cs.bgu.ac.il

Patrick Prosser
patrick.prosser@gla.ac.uk

Peter J. Stuckey
pstuckey@unimelb.edu.au

[1] Department of Computer Science, Ben Gurion University of the Negev, Negev, Israel

[2] School of Computing Science, University of Glasgow, Glasgow, UK

[3] School of Computing and Information Systems, University of Melbourne, Melbourne, Australia

1 Introduction

The canonical graph representation problem is pertinent to a wide range of scientific applications. It is closely related to the graph isomorphism problem, as two graphs are isomorphic if and only if they have the same canonical representation. Examples of applications where graphs (and their representations) have been used include data-mining [1], mathematical chemistry [2, 3], computer vision [4], and bioinformatics [5]. The two problems are polytime equivalent, and are among the few that are known to be in NP but not known either to be solvable in polynomial time, nor to be NP-complete. Recently László Babai has claimed to have produced a quasi-polynomial time algorithm for graph isomorphism. This result is yet to be published.

There are a variety of software tools devoted to solving the two problems "in practice," one of which is `nauty` (No AUTomporphisms Yes?), due to McKay [6, 7]. `Nauty` is sometimes referred to as the world's fastest isomorphism testing program. It is also able to produce a canonically-labeled isomorph of a graph to assist in isomorphism testing. The nauty package includes a suite of programs called `gtools` which are useful for processing files of graphs stored in one of two compressed formats. In this paper we make use of the `shortg` tool to remove isomorphs from a file of graphs stored in the `graph6` format.

This paper is about constraint problems which involve the search for either a single graph that satisfies certain properties, or all graphs that do so. For example, consider the problem to determine if there exists an undirected graph with 31 vertices, 81 edges, and which does not contain cycles of length 4 or less. This question arises in "extremal graph theory" [8], and its answer was unknown prior to the work described in this paper [9]. The search space for problems of this type is enormous, and search may be optimized by restricting it to focus on canonical representations, or to avoid isomorphic graphs as much as possible. The general idea is to "break" symmetries in the search space that derive from the fact that the actual names of the vertices in a solution do not matter. When searching for a graph coloring where the edges of the graph are associated with colors, on top of graph isomorphism, solutions are typically closed under permutations of the colors: the actual names of the colors do not matter. However, it is not clear how to apply this idea when searching for a graph. In this type of problem the graph is a variable, so graph algorithms for canonical representation and isomorphism, as well as tools such as `nauty`, all of which operate on given graphs, do not apply. This paper provides a solution to this problem.

We assume a setting where testing for the existence of a graph G satisfying a property P is posed as a Boolean constraint $P(A_G)$ on the variables of the Boolean adjacency matrix A_G of G. We follow the approach advocated by Crawford et al. [10], where a predicate, $\mathsf{sb}(A_G)$, is introduced to break symmetries in the search space. In this way the satisfiability of $P(A_G)$ is equivalent to that of $P(A_G) \wedge \mathsf{sb}(A_G)$. Ideally, $\mathsf{sb}(A_G)$ is satisfied by a single member of each equivalence class of A_G under graph isomorphism, thus drastically restricting the search space for $P(A_G) \wedge \mathsf{sb}(A_G)$. However, this is not realistically possible as such a predicate also determines a canonical representation. In practice, it is sufficient that $\mathsf{sb}(A_G)$ is satisfied by at least one member of the equivalence class of A_G under isomorphism (typically by more than one) and in this case we say that sb is a *symmetry breaking predicate*. Shlyakhter [11] notes that the difficulty is to identify a symmetry-breaking predicate which is both *effective* (rules out a large portion of the search space) and *compact* (so that checking the additional constraints does not prohibitively slow down the search).

The presentations in [10, 11] consider symmetry breaking in terms of isomorphism, but focus on different structures such as acyclic digraphs, relations, permutations and functions. We introduce a novel, effective and compact predicate to break symmetries on graph representation. We consider two different scenarios where an (unknown) graph G is represented by its adjacency matrix A_G and the elements at positions i, j in A_G are: Boolean variables indicating the presence of an edge between vertices i and j; and (2) integer variables from a finite domain $C = \{0, 1, \ldots, k\}$ indicating the presence of a colored edge between vertices i and j where a value $c > 0$ indicates a c-colored edge.

This paper extends preliminary results presented in [12], in which our symmetry breaking constraints were introduced and applied to solve several open instances of a problem in extremal graph theory regarding the maximal number of edges in a graph with v vertices and no cycles of length k or less (i.e. graphs with girth at least $k + 1$), for $k = 4$ and $k = 5$. We also use our approach to determine the number of non-isomorphic extremal graphs of girth at least 5. Here, we exploit some properties of extremal graphs (presented and proved in [13]) which then enable us to apply our approach to solve additional open instances of maximal graphs with girth 5. We also extend the approach to apply to graph coloring problems and demonstrate its impact on the search for Ramsey numbers and colorings [14]. Note that in [15] we apply the results of this paper to determine the precise value of $R(4, 3, 3)$. This was an open problem for over 30 years.

2 Graphs and their canonical representation

Throughout this paper we consider undirected simple graphs without loops or multiple edges. We focus on finite graphs and typically name the n vertices of a graph in the set $\{1, \ldots, n\}$. We denote the Boolean values *true* and *false* by 1 and 0 respectively.

Definition 1 (Graph) A *graph* $G = (V, E)$ has vertices $V = \{1, \ldots, n\}$ and edges $E \subseteq V \times V$ where $(x, y) \in E \Rightarrow (y, x) \in E$. The Boolean *adjacency matrix*, A_G of G, is the $n \times n$ symmetric matrix where $A_G[x, y] \Leftrightarrow (x, y) \in E$. The i^{th} row of matrix A is denoted by $A[i]$, and $A[i, j]$ denotes the j^{th} element of $A[i]$. The degree of vertex $u \in V$ is $degree(u) = |\{(u, v)|(u, v) \in E\}|$. We denote the minimum and maximum degrees of the vertices in G as $\delta(G)$ and $\Delta(G)$, or δ and Δ when the context is clear.

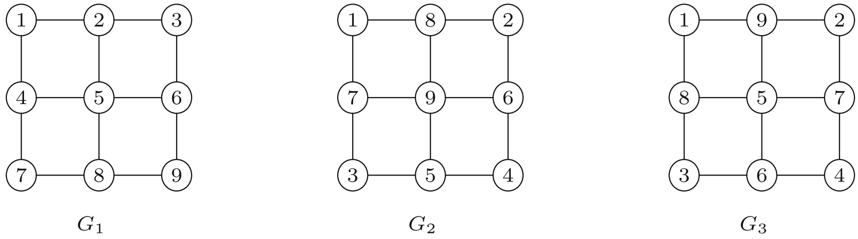
In Section 5, we extend Definition 1 to colored graphs where the edges are associated with colors from a given finite domain of positive integers.

Example 1 Figure 1 illustrates three graphs with corresponding adjacency matrices.

We use cycle notation to represent permutations. For example, the permutation (1,2,6)(3,4,5) on set $\{1, \ldots, 7\}$ maps 1 to 2, 2 to 6, 6 to 1, and 3 to 4, 4 to 5, and 5 to 3 (cycles of length 1 are not included in the representation).

Definition 2 (permuting vertices) Let $G = (V, E)$ be a graph with n vertices, A_G the adjacency matrix for G, and π a permutation on $\{1, \ldots, n\}$. Then $\pi(G)$ is the graph obtained by permuting the vertices of G using π. Formally, $\pi(G) = (V, E')$ where $E' = \{(\pi(x), \pi(y))|(x, y) \in E\}$ and $\pi(A_G)$ is the adjacency matrix of $\pi(G)$.

Definition 3 (graph isomorphism) G and G' are isomorphic if there exists a permutation π such that $A_G = \pi(A_{G'})$.



$$A_{G_1} = \begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
2 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
4 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
5 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
6 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
8 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
9 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0
\end{array}$$

$$A_{G_2} = \begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
5 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
6 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
7 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
8 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
9 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0
\end{array}$$

$$A_{G_3} = \begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
4 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
6 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
7 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
8 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
9 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}$$

Fig. 1 Three example graphs and their adjacency matrices

Example 2 The graphs in Fig. 1 are isomorphic. We can permute G_1 to G_2 using $\pi_1 = (2, 8, 5, 9, 4, 7, 3)$ and G_1 to G_3 using $\pi_2 = (2, 9, 4, 8, 6, 7, 3)$.

Because of the way they are presented, the graphs in Example 2 (Fig. 1) are very obviously isomorphic. However isomorphism (particularly for large graphs) is not usually so easy to detect. For example, the graphs in Fig. 2 are isomorphic (using $\pi = (1, 2, 3)(4, 5, 6)(7, 10, 11)$) but, because of the placement of the vertices, are not obviously so. The fact that the two graphs are isomorphic can be ascertained from their adjacency matrices using a graph-theoretic tool such as nauty [6, 7].

Definition 4 (sequences, lexicographic order) Let A be matrix and $A[i]A[j]$ the concatenation of rows i and j (viewed as sequences). The length of a sequence s is denoted $|s|$.

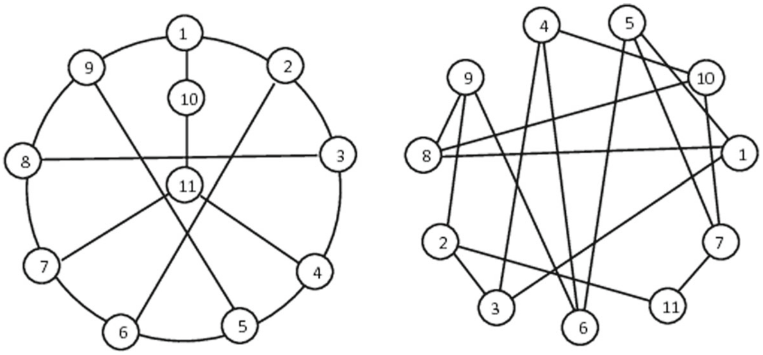


Fig. 2 Isomorphic graphs presented with isomorphism concealed

We use $\preceq$ to denote the usual lexicographic order on sequences. We extend this notation in the obvious way: for matrices, with n and m rows respectively, $A \preceq B$, if and only if $A[1]A[2]\cdots A[n] \preceq B[1]B[2]\ldots B[m]$; and for graphs, $G \preceq G'$ if and only if $A_G \preceq A_{G'}$.

One way to define a canonical representation of a graph is to take the smallest graph (i.e. in the lexicographic order) which is isomorphic to G [16]. This is the definition which we adopt throughout the paper.

Definition 5 (canonical form of a graph) The canonical form of a graph G is the graph with $can(G) = \min_{\preceq}\{\pi(G)|\pi \text{ is a permutation}\}$. We say that G is canonical if $G = can(G)$.

Example 3 Consider the graphs of Fig. 1. The graph G_3 is the canonical representation of G_1, G_2 and G_3.

Note that the canonical representation of a graph does not necessarily order the vertices by degree. In Fig. 1, the vertices of G_2 are ordered by degree: vertices $\{1, 2, 3, 4\}$ are of degree 2, vertices $\{5, 6, 7, 8\}$ are of degree 3 and vertex 9 is degree 4. But this is not the case for the canonical form, G_3.

3 Symmetry breaking on representation

We first consider a symmetry breaking predicate, introduced without proof in [17], which constrains the rows of an adjacency matrix to be sorted lexicographically in non-decreasing order.

Definition 6 (lexicographic symmetry break) Let A be an $n \times n$ adjacency matrix. We define

$$\mathsf{sb}_\ell(A) = \bigwedge_{i=1}^{n-1} A[i] \preceq A[i+1]$$

Observe the graphs in Fig. 1. We have $\mathsf{sb}_\ell(A_{G_1}) = \mathit{false}$, $\mathsf{sb}_\ell(A_{G_2}) = \mathit{false}$, and $\mathsf{sb}_\ell(A_{G_3}) = \mathit{true}$.

Definition 6 is more subtle than might first appear. It defines a symmetry breaking predicate only because for every adjacency matrix A, $\mathsf{sb}_\ell(A')$ is true for at least one of the matrices A' isomorphic to A. Reversing the order, i.e. insisting that $A[i] \succeq A[i+1]$, would not define a symmetry breaking constraint. Consider for example any representation of the graph G with 2 vertices and a single edge. Then $A_G[1] \not\succeq A_G[2]$. The subtlety arises because, in contrast to the case of breaking symmetries in matrix problems where rows and columns can be reordered, such as in [18–20], here we need to reorder rows and columns simultaneously, both in the same way. To prove the correctness of Definition 6 it is sufficient to show that $\mathsf{sb}_\ell(can(A))$ holds.

Theorem 1 *Let G be a graph. Then $\mathsf{sb}_\ell(can(A_G))$.*

Proof Let A be canonical and assume to the contrary that A does not satisfy $\mathsf{sb}_\ell(A)$. Let i be such that $A[i] \not\preceq A[i+1]$. It follows that there is a j such that for every $1 \leq j' < j$,

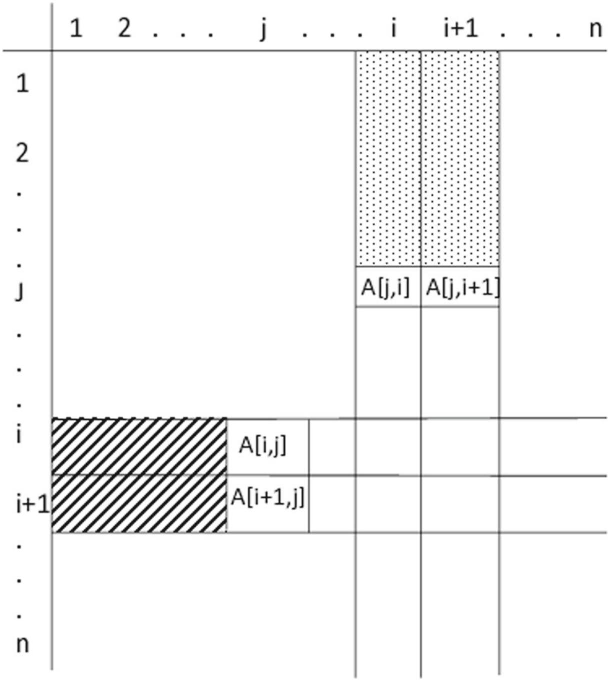


Fig. 3 Graph A, Theorem 1a, $j < i$. Similarly shaded sections of rows/columns i and $i + 1$ are identical

$A[i, j'] = A[i+1, j']$ and $A[i, j] > A[i+1, j]$. Let B be the matrix obtained by swapping rows $i, i + 1$ as well as columns $i, i + 1$. We show that $B \prec A$ in contradiction to A being canonical. Since $A[i, j] > 0$, $i \neq j$ and there are two cases to consider.

(a) $j < i$: Since $j - 1$ length prefixes of $A[i]$ and $A[i + 1]$ are equal (the diagonally striped regions in Fig. 3), $A[i, 1] \cdots A[i, j - 1] = B[i, 1] \cdots B[i, j - 1]$. Note also that $A[i'] = B[i']$ for $1 \leq i' < j$. This is because the only elements to be swapped in these rows are those in columns i and $i + 1$ (the dotted regions in Fig. 3). These elements are equal because A is symmetric and $A[i, i'] = A[i+1, i']$ for $i < j$. Hence the first cell to differ in A and B is at position $[j, i]$, and, since $A[j, i] > A[j, i + 1]$, $A[j, i] > B[j, i]$. So $B \prec A$. Contradiction.

(b) $j > i$: By a similar argument to the above, the $A[i, 1] \cdots A[i, j - 1] = B[i, 1] \cdots B[i, j - 1]$ and $A[i'] = B[i']$ for $1 \leq i' < i$ (so the similarly shaded sections of rows i and $i + 1$ and columns i and $i + 1$ in Fig. 4 are identical). It follows that the first cell to differ in A and B is at position $[i, j]$ and that $A[i, j] > B[i, j]$. So $B \prec A$. Contradiction. □

We now proceed to strengthen this notion of symmetry breaking. The following example illustrates a symmetry not captured by $\mathsf{sb}_\ell(A)$.

Example 4 Consider the adjacency matrix A_1 depicted in Fig. 5 for which $\mathsf{sb}_\ell(A_1) = \mathit{true}$ as the rows are ordered lexicographically. Observe that $A_1[2] \preceq A_1[3]$ independent of whether we swap the vertices (rows and columns) 2 and 3, or not. Adjacency matrix A_2 depicted in Fig. 5 is the result of this swap and it too satisfies $\mathsf{sb}_\ell(A_2) = \mathit{true}$. However, it

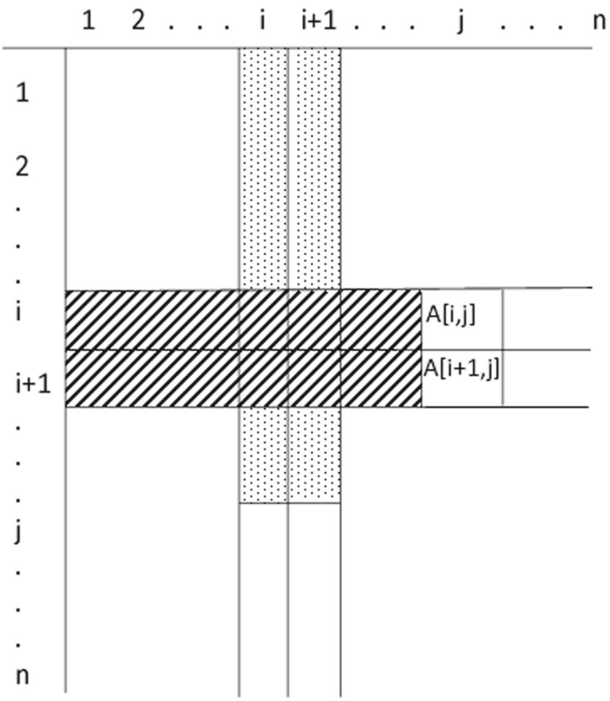


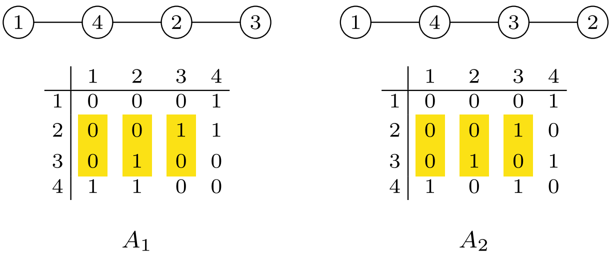
Fig. 4 Graph A, Theorem 1(b), $j > i$. Similarly shaded sections of rows/columns i and $i + 1$ are identical

is "closer" to canonical as $A_2 \preceq A_1$. Indeed A_2 is the canonical representative of this graph. Figure 5 highlights that the first 3 elements of rows 2 and 3 are invariant under vertex swap.

In view of Example 4 we introduce the following definition and then introduce a stronger symmetry breaking constraint.

Definition 7 (extended lexicographic order)

Let s be a sequence and $I \subseteq \{1, \ldots, |s|\}$. We denote by $(s \restriction I)$ the sequence obtained from s by simultaneously omitting the elements at positions I. For a set of natural numbers I we denote by $\preceq_I$ the order on sequences of length at least $\max(I)$ defined by: $s_1 \preceq_I s_2 \Leftrightarrow (s_1 \restriction I) \preceq (s_2 \restriction I)$.



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 1 \\ 3 & 0 & 1 & 0 & 0 \\ 4 & 1 & 1 & 0 & 0 \end{array}$$

A_1

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 1 \\ 4 & 1 & 0 & 1 & 0 \end{array}$$

A_2

Fig. 5 Graphs and adjacency matrices for Example 4

Definition 8 (improved lexicographic symmetry break) Let A be an $n \times n$ adjacency matrix. We define

$$\mathsf{sb}^*_\ell(A) = \bigwedge_{i<j} A[i] \preceq_{\{i,j\}} A[j]$$

Theorem 2 *If A is an $n \times n$ adjacency matrix and $\mathsf{sb}^*_\ell(A)$ then $\mathsf{sb}_\ell(A)$.*

Proof Suppose that $\mathsf{sb}^*_\ell(A)$ and that for some row i, $A[i] \not\preceq A[i+1]$. Since $A[i] \preceq_{\{i,i+1\}} A[i+1]$, we must have that $A[i,1] \cdots A[i,i-1] = A[i+1,1] \cdots A[i+1,i-1]$ and $A[i,i+2] \cdots A[i,n] \preceq A[i+1,i+2] \cdots A[i+1,n]$. Since $A[i,i] = 0$ and $A[i,i+1] = A[i+1,i]$ either $A[i,i] < A[i+1,i]$ or $A[i,i+1] = A[i+1,i] = 0$. In both cases $A[i] \preceq A[i+1]$, and we have a contradiction. □

Observe that Definition 8 introduces $O(n^2)$ constraints on lexicographic order whereas Definition 6 introduces only $O(n)$. This is needed because we lack a "transitivity" like property stating that if $s_1 \preceq_{\{i,j\}} s_2$ and $s_2 \preceq_{\{j,k\}} s_3$ then also $s_1 \preceq_{\{i,k\}} s_3$. The fact that no such property holds is illustrated by the following example.

Example 5 Consider the adjacency matrix A shown in Fig. 6. While clearly $A[1] \preceq_{\{1,2\}} A[2]$ and $A[2] \preceq_{\{2,4\}} A[4]$, it is not the case that $A[1] \preceq_{\{1,4\}} A[4]$.

Interestingly, transitivity does hold for rows at a distance of two apart.

Theorem 3 $A[i] \preceq_{\{i,i+1\}} A[i+1] \wedge A[i+1] \preceq_{\{i+1,i+2\}} A[i+2] \Rightarrow A[i] \preceq_{\{i,i+2\}} A[i+2]$

Proof Assume the premise and adopt the following representation where the boxed elements are at positions i, $i+1$ and $i+2$ in the sequences.

$$\begin{aligned} A[i] &= S_1 \boxed{0\ x\ y} T_1 \\ A[i+1] &= S_2 \boxed{x\ 0\ z} T_2 \\ A[i+2] &= S_3 \boxed{y\ z\ 0} T_3 \end{aligned}$$

From the premise and by definition of $\preceq_I$, we have $S_1 y T_1 \preceq S_2 z T_2$ and $S_2 x T_2 \preceq S_3 y T_3$. We prove that $S_1 x T_1 \preceq S_3 z T_3$ which gives the result. There are two cases: either (a) $S_1 \prec S_2$, and since $S_2 \preceq S_3$ we have $S_1 \prec S_3$ and the result holds; or (b) $S_1 = S_2$, then either $S_2 \prec S_3$, and the result follows, or $S_2 = S_3$, and it remains to show that $x T_1 \preceq z T_3$. Suppose $y \prec z$ then since $x \preceq y$ we have that $x \prec z$ and the result holds. Otherwise, $y = z$. Suppose $x \prec y$ then clearly $x \prec z$ and the result holds. So assume $x = y = z$. Then we have that $T_1 \preceq T_2$ and $T_2 \preceq T_3$ and hence the result holds. □

Intuitively, transitivity fails for the general case (for rows i, j and k) due to the possibility that the position of the first element in row i that is not equal to the corresponding element in row j may be between i and j or between j and k (or a similar scenario for rows j and k or

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & \colorbox{yellow}{0} & \colorbox{yellow}{1} & \colorbox{yellow}{1} \\ 2 & 0 & 0 & \colorbox{yellow}{1} & \colorbox{yellow}{0} & \colorbox{yellow}{0} \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 & 0 \end{array} \qquad \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 \\ 2 & \colorbox{yellow}{0} & 0 & \colorbox{yellow}{1} & 0 & \colorbox{yellow}{0} \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & \colorbox{yellow}{1} & 0 & \colorbox{yellow}{0} & 0 & \colorbox{yellow}{0} \\ 5 & 1 & 0 & 0 & 0 & 0 \end{array} \qquad \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & \colorbox{yellow}{0} & \colorbox{yellow}{0} & 1 & \colorbox{yellow}{1} \\ 2 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & 1 & \colorbox{yellow}{0} & \colorbox{yellow}{0} & 0 & \colorbox{yellow}{0} \\ 5 & 1 & 0 & 0 & 0 & 0 \end{array}$$

$$A[1] \preceq_{\{1,2\}} A[2] \qquad A[2] \preceq_{\{2,4\}} A[4] \qquad A[1] \not\preceq_{\{1,4\}} A[4]$$

Fig. 6 Extended lexicographic comparisons for Example 5 where only the highlighted entries are compared

rows i and k), which is clearly not possible for the case in Theorem 3. Using a representation similar to the above, where the boxed elements are in positions i, j and k in the sequences:

$$A[i] = S_1 \boxed{0} T_1 \boxed{x} U_1 \boxed{y} V_1$$
$$A[j] = S_2 \boxed{x} T_2 \boxed{0} U_2 \boxed{z} V_2$$
$$A[k] = S_3 \boxed{y} T_3 \boxed{z} U_3 \boxed{0} V_3$$

If $S_1 = S_2 = S_3$, $T_3 \prec T_1 \prec T_2$, $x = 0$ and $y = 1$ then $S_1T_1U_1yV_1 \preceq S_2T_2U_2zV_2$ and $S_2xT_2U_2V_2 \preceq S_3yT_3U_3V_3$, i.e. $A[i] \preceq_{\{i,j\}} A[j]$ and $A[j] \preceq_{\{j,k\}} A[k]$, but $S_1T_1xU_1V_1 \not\preceq S_3T_3zU_3V_3$, i.e $A[i] \not\preceq_{\{i,k\}} A[k]$.

By removing the redundancy implied by Theorem 3 we can remove $O(n)$ constraints and refine Definition 8 thus:

Corollary 1

$$\mathsf{sb}^*_\ell(A) = \bigwedge_{\substack{i<j \\ j-i \neq 2}} A[i] \preceq_{\{i,j\}} A[j]$$

The following proves that sb^*_ℓ is a symmetry-breaking predicate.

Theorem 4 *Let A be a canonical adjacency matrix. Then $\mathsf{sb}^*_\ell(A)$ holds.*

Proof Let A be the canonical adjacency matrix for a graph G and assume to the contrary that A does not satisfy $\mathsf{sb}^*_\ell(A)$. That is, there exist i and j such that $i < j$ and $A[i] \not\preceq_{\{i,j\}} A[j]$. Let π denote the permutation which swaps vertices i and j in G. We show that $B = A_{\pi(G)} \prec A$. Let $k \neq i, j$ be the first column at which $A[i]$ and $A[j]$ differ (except possibly for columns i and/or j). It follows that for every $1 \leq k' < k$, $k \neq i, j$, $A[i, k'] = A[j, k']$ and $A[i, k] > A[j, k]$. There are 3 cases to consider, $k < i$, $i < k < j$ and $k > j$. We consider these cases below and illustrate them using Figs. 7, 8 and 9. In each case similarly shaded sections of rows/columns i and j are identical, and the grey squares denote the unknown (but identical) values $A[i, j]$ and $A[j, i]$.

(a) $k < i$: Since $k - 1$ length prefixes of $A[i]$ and $A[j]$ are equal (the diagonally striped regions in Fig. 7), $A[i, 1] \cdots A[i, k-1] = B[j, 1] \cdots B[j, k-1]$. Note also that $A[i'] = B[i']$ for $1 \leq i' < k$ (the dotted regions in Fig. 7). The i and j elements in $A[i']$ are equal because A is symmetric and $A[i, i'] = A[j, i']$ for $i' < k$. Hence the

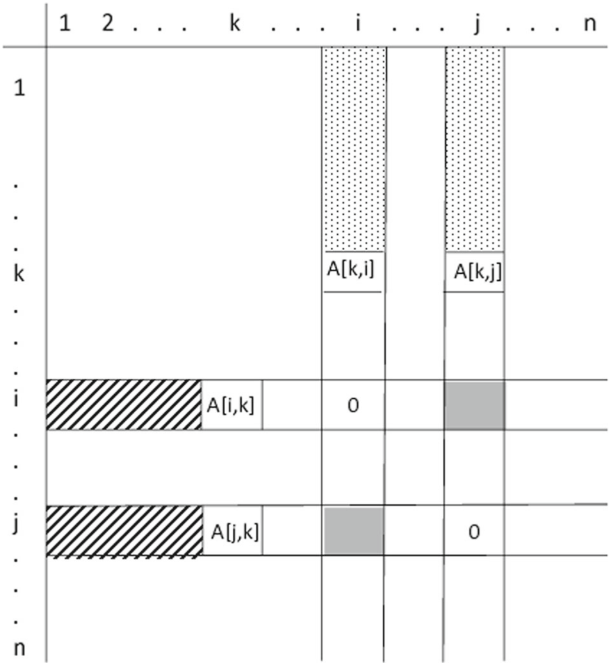


Fig. 7 Graph A, Theorem 4(a), $k < i$. Similarly shaded sections of rows/columns i and j are identical and grey shaded squares are unknown but identical values

first cell to differ in A and B is at position $[k, i]$, and $A[k, i] > B[k, i]$. So $B \prec A$. Contradiction.

(b) $i < k < j$: By a similar argument to the above, $A[i, 1] \cdots A[i, k-1] = A[j, 1] \cdots A[j, k-1]$, except possibly in column i (the diagonally striped regions in Fig. 8). By symmetry $A[1, i] \cdots A[k-1, i] = B[j, 1] \cdots B[j, k-1]$, except possibly in row i (the dotted regions in Fig. 8) and so, in particular, for $i' < i$, $A[i'] = B[i']$. Since $A[i, i] = 0$ and A is canonical, $A[j, i]$ must be 0. Hence, for $k' < k$, $A[i, k'] = B[i, k']$. It follows that the first cell to differ in A and B is at position $[i, k]$ and that $A[i, k] > B[i, k]$.

(c) $k > j$: By a similar argument to the above, $A[i, 1] \cdots A[i, k-1] = A[j, 1] \cdots A[j, k-1]$, except possibly in columns i and j (the diagonally striped regions in Fig. 9) and $A[1, i] \cdots A[k-1, i] = B[1, j] \cdots B[k-1, j]$, except possibly in rows i and j (the dotted regions in Fig. 9). In particular, for $i' < i$, $A[i'] = B[i']$. By a similar argument to part (b), as $A[i, i] = 0$ and A is canonical, $A[j, i]$ must be 0. Since $A[j, i] = A[i, j]$ (by symmetry) and $A[j, j] = 0$ and we already know that rows i and j are identical up to column $k-1$ otherwise, it follows that the first cell to differ in A and B is at position $[i, k]$ and that $A[i, k] > B[i, k]$.

□

Another way to think about $\mathsf{sb}^*_\ell(A)$ is that it prevents us from creating a form of the graph where swapping any two rows will lead to a lexicographically smaller graph.

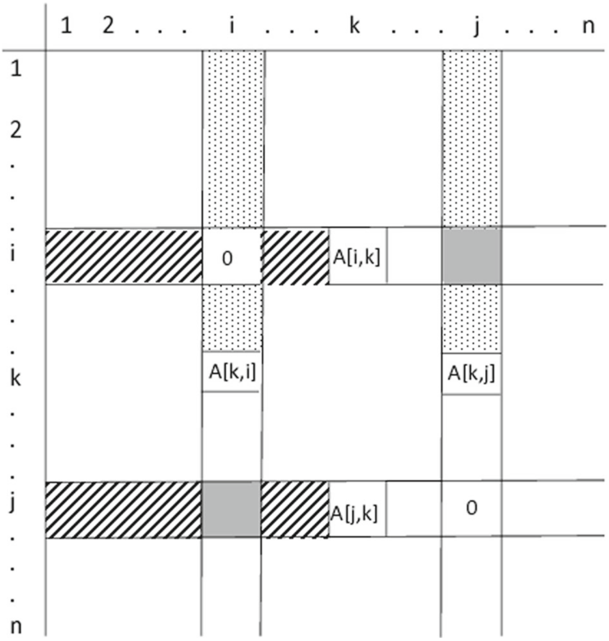


Fig. 8 Graph A, Theorem 4(b), $i < k < j$. Similarly shaded sections of rows/columns i and j are identical and grey shaded squares are unknown but identical values

Theorem 5 *Let A be an adjacency matrix for graph G where $\mathsf{sb}^*_\ell(A)$ holds. Let π be a permutation that swaps vertices i and j in G. Then $A \preceq A_{\pi(G)}$.*

Proof Let $B = A_{\pi(G)}$. Assume w.l.o.g. that $i < j$. Suppose to the contrary that $B \prec A$ we show that $\mathsf{sb}^*_\ell(A)$ does not hold.

Let $A[i] = S_1\textcircled{0}S_2\textcircled{x}S_3$ and $A[j] = T_1\textcircled{x}T_2\textcircled{0}T_3$ where the circled entries occur at positions i and j in the sequences. Now we know that $B[i] = T_1\textcircled{0}T_2\textcircled{x}T_3$ and $B[j] = S_1\textcircled{x}S_2\textcircled{0}S_3$ by the definition of B.

Arrays A and B are shown in Fig. 10. We also label the regions of columns A and B that are (by symmetry) transpose of one of $S_1, \cdots, S_3$ or $T_1, \cdots, T_3$.

Suppose the first position where B and A differ is row k column l. Then $B[k] \prec A[k]$. By the nature of B, either k or l is i or j. Suppose that $l = i$ or $l = j$ then, since $A[k,i] \neq B[k,j]$ if and only if $A[k,j] \neq B[k,j]$, we can assume that $l = i$. Similarly, if k is i or j we can assume that $k = i$. Also, by symmetry, if elements in row i' and column j' differ in A and B, so do the elements in row j' and column i'. We can therefore assume that the first element in A that differs from the corresponding element in B is one of: $k < i, l = i$; $k = i, i < l < j$ or $k = i, l > j$, i.e. in one of the shaded areas in Fig. 10.

If $k < i$, $l = i$, then since $B[k] \prec A[k]$, $B[k,i] < A[k,i]$. As $B[k,i] = A[k,j]$ we have $A[k,j] < A[k,i]$ and, by symmetry, $A[j,k] < A[i,k]$. Since k is the first position in which $A[i]$ and $A[j]$ differ, $A[i] \not\preceq_{\{i,j\}} A[j]$.

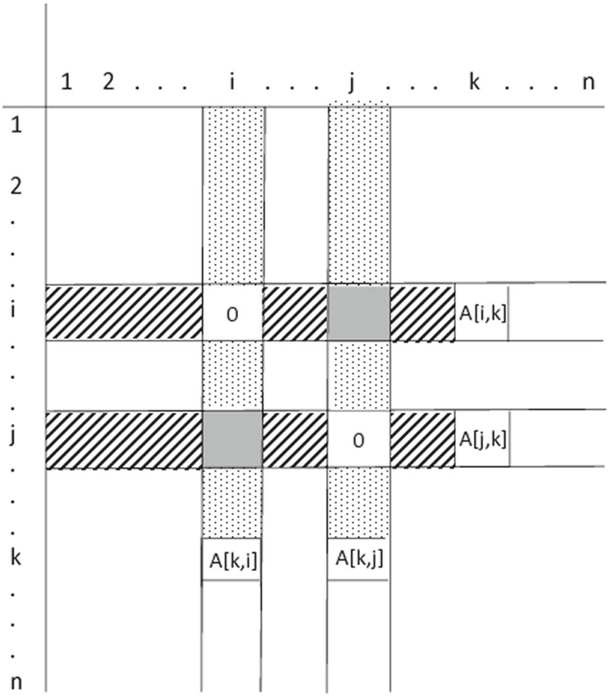


Fig. 9 Graph A, Theorem 4(c), $j < k$. Similarly shaded sections of rows/columns i and j are identical and grey shaded squares are unknown but identical values

If $k = i$, and $i < l < j$ or $l > j$ then since $B[k] \prec A[k]$, $B[i,l] < A[i,l]$. As $B[i,l] = A[j,l]$ we have $A[j,l] < A[i,l]$. Since l is the first position in which $A[i]$ and $A[j]$ differ, $A[i] \not\preceq_{\{i,j\}} A[j]$. □

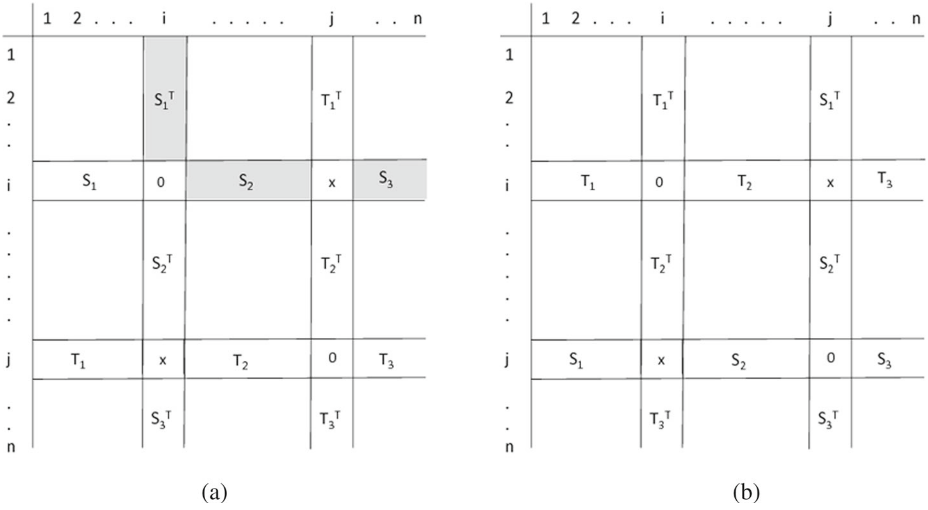


Fig. 10 Graphs A and B, Theorem 5. Shaded areas in (**a**) denote the possible position of the first element to differ in (**a**) and (**b**)

Note that often we may wish to separate vertices of the graph into equivalence classes a priori, and generate a graph that satisfies those equivalence classes. We can still use (extended) lexicographic ordering to help constrain the resulting adjacency matrices, since we can extend Theorem 4 to this case.

Definition 9 (ordered partition)] Let G be a graph. Then $P = \{P_1, \ldots, P_p\}$ is an *ordered partition* of the vertices of G if $\forall 1 \leq i < j \leq p, v_i \in P_i \wedge v_j \in P_j \Rightarrow v_i < v_j$.

Definition 10 (partition preserving permutation) Let $P = \{P_1, \ldots, P_p\}$ be an ordered partition on the vertices of G. A permutation π on the vertices of G is *partition preserving* for P if $\forall 1 \leq i \leq p, \forall v_i \in P_i, \pi(v_i) \in P_i$.

Example 6 Consider the graph G_2 from Fig. 1 and the ordered partition $P = \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \{9\}\}$, which partitions vertices by degree. Then the permutation $\pi = (2, 3, 4)$ is partition preserving for P. It maps elements in P_1 to other elements in P_1 and fixes elements in the other parts.

Definition 11 (canonical partitioned adjacency matrix) The canonical form of a graph G with respect to an ordered partition P is the graph $can(G, P) = \min_{\preceq}\{\pi(G) | \pi$ is a partition preserving permutation for $P\}$. We say that G is canonical for P if $G = can(G, P)$.

We can define a symmetry breaking predicate for partitioned graphs as follows:

Definition 12 (partitioned lexicographic symmetry break) Let A be an $n \times n$ adjacency matrix and $P = \{P_1, P_2, \ldots, P_p\}$ be an ordered partition. We define

$$\mathsf{sb}^*_\ell(A, P) = \bigwedge_{k=1}^{p} \bigwedge_{\substack{\{i,j\} \subseteq P_k, i < j \\ j - i \neq 2}} A[i] \preceq_{\{i,j\}} A[j]$$

Theorem 6 *Let G be a canonical partitioned graph for an ordered partition P. Then $\mathsf{sb}^*_\ell(A_G, P)$ holds.*

Proof Let A be the canonical adjacency matrix for graph G and assume to the contrary that A does not satisfy $\mathsf{sb}^*_\ell(A, P)$. That is, there exists a partition P_k and $\{i, j\} \subseteq P_k$ with $i < j$ where $A[i] \npreceq_{\{i,j\}} A[j]$. Let $B = A_{\pi(G)}$ where π swaps i and j. Note that π is a partition preserving permutation for P. Using a proof similar to that of Theorem 4 we can show that $B \prec A$. □

Example 7 Let G be the graph G_2 from Fig. 1 and the ordered partition $P = \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \{9\}\}$, which partitions vertices by degree. Then G is canonical for P (even though, as shown in Example 3, G is not canonical). Clearly $\mathsf{sb}^*_\ell(A_G, P)$ holds, although $\mathsf{sb}^*_\ell(A_G)$ does not.

Note that we exploit the fact that symmetry breaking can be applied without disturbing a defined partitioning in Section 4 when enforcing the existence of embedded stars (but not otherwise).

4 Extremal graph problems

Extremal graph theory [8] is the study of graphs that are maximal (or minimal) in some way (for example in terms of number of edges) and which satisfy a given property. Extremal graph theory has many applications both in other areas of mathematics and fields including, for example, chemistry [21], biology [22] and cryptography [23].

We apply a constraint-based approach to some extremal graph problems and illustrate the advantage of symmetry breaking on the graph representation.

The girth of a graph is the size of the smallest cycle contained in it. Let $\mathcal{F}_k(v)$ denote the set of graphs with v vertices and girth at least $k+1$. Let $f_k(v)$ denote the maximum number of edges in a graph in $\mathcal{F}_k(v)$. A graph in $\mathcal{F}_k(v)$ with $f_k(v)$ edges is called *extremal*. The number of non-isomorphic extremal graphs in $\mathcal{F}_k(v)$ is denoted $F_k(v)$. Extremal graph problems involve discovering values of $f_k(v)$ and $F_k(v)$ and finding witnesses. In [24] the authors attribute the discovery of values $f_4(v)$ for $v \leq 24$ to [9] and for $25 \leq v \leq 30$ to [25]. Hand proofs for $f_4(v)$ for $40 \leq v \leq 49$ are presented in [26]. In [9] the authors report values of $F_4(v)$ for $v \leq 21$. In [9] and [27] algorithms are applied to compute lower bounds on $f_4(v)$ for $31 \leq v \leq 200$. Some of these lower bounds are improved in [28] and improved upper bounds for $33 \leq v \leq 42$ are proved in [29]. Currently available values of $f_4(v)$ and of $F_4(v)$ are available as sequences `A006856` and `A159847` of the On-Line Encyclopedia of Integer Sequences [30].

Our basic constraint model is shown in Fig. 11 where we assume given values of v (number of vertices) and e (number of edges) and that A is a $v \times v$ matrix of Boolean variables. Constraint (1) states that the graph is simple (symmetric with no self loops), Constraints (2) and (3) express that there are no cycles of length 3 or 4, and Constraint (4), that the number of edges is e. Constraints (2) and (3) are implemented more efficiently. We introduce additional Boolean variables for each triplet of (distinct) vertices i, j, k with $i < k$: $x_{i,j,k} \Leftrightarrow A[i,j] \wedge A[j,k]$ represents a length 2 path between i and k via j; and $x_{i,k} \Leftrightarrow \vee\{x_{i,j,k} | j \neq i, j \neq k\}$ represents the existence of any length 2 path between i and k. We then express Constraints (2) and (3) as $\forall_{i,k}.A[i,k] + x_{i,k} < 2$ and $\forall_{i,k}. \sum_j x_{i,j,k} < 2$.

$$\forall_{1 \leq i < j \leq v}.\ (A[i,j] = A[j,i] \quad \text{and} \quad A[i,i] = 0)$$

$$\forall_{i,j,k}.\ A[i,j] + A[j,k] + A[k,i] < 3$$

$$\forall_{i,j,k,l}.\ A[i,j] + A[j,k] + A[k,l] + A[l,i] < 4$$

$$\sum_{1 \leq i < j \leq v} A[i,j] = e$$

$$\forall_{1 \leq i \leq v}.\delta \leq \sum A[i] \leq \varDelta,$$

$$\delta = \min_i(\sum A[i])$$

$$\varDelta = \max_i(\sum A[i])$$

Fig. 11 Basic constraint model for extremal graph problems (no cycles of length 4 or less) with v vertices and e edges

To explain Constraints (5)–(7) we recall Propositions 2.6 and 2.7 from [9] which state that for every graph in $\mathcal{F}_4(v)$ with e edges the minimum and maximum vertex degrees, denoted δ and Δ, satisfy the following equations (assuming $v \geq 1$):

$$\begin{gathered} v \geq 1 + \Delta\delta \geq 1 + \delta^2, \text{ and} \\ \delta \geq e - f_4(v-1), \text{ and } \Delta \geq \lceil 2e/v \rceil \end{gathered} \tag{1}$$

Given values for v and e we model the problem separately for each potential pair (δ, Δ) introducing constraints (5) to (7). In addition to the above constraints we introduce symmetry breaking constraints sb_ℓ or sb^*_ℓ.

Example 8 For $v = 31$ and $e = 80$ the possible (δ, Δ) pairs satisfying Eq. 1 are $\{(4,6), (4,7), (5,6)\}$. Similarly, for $v = 31$ and $e = 81$ there is a single pair, $(5,6)$.

We describe three experiments to evaluate the impact of different symmetry breaking strategies. Experiments were run using the BEE [31] constraint solver.

Note that for $v < 20$ our experiments often take longer to run (even with sb^*_ℓ) than using nauty's geng tool [6] – for example we take 4.7 seconds to find $F_4(15)$, whereas geng takes only 1.8 seconds. However, for larger values of v, geng can not find a solution within

Table 1 Computing $f_4(v)$ (time in seconds; timeout 4hrs)

		No symmetry break		sb_ℓ		sb^*_ℓ		sb_ℓ + star	
v	f$_4$(v)	sat	unsat	sat	unsat	sat	unsat	sat	unsat
11	16	0.22	∞	0.15	0.30	0.10	0.22	0.07	0.04
12	18	0.11	0.00	0.10	0.00	0.11	0.00	0.05	0.00
13	21	0.09	0.00	0.13	0.00	0.15	0.00	0.04	0.00
14	23	0.13	∞	0.23	1.14	0.20	0.58	0.13	0.08
15	26	0.16	0.00	0.27	0.00	0.30	0.00	0.11	0.00
16	28	0.18	∞	0.30	87.18	0.61	32.20	0.25	3.63
17	31	0.28	0.00	0.31	0.00	0.48	0.00	0.19	0.00
18	34	0.29	0.00	0.40	0.00	0.60	0.00	0.27	0.00
19	38	0.34	0.00	0.41	0.00	0.57	0.00	0.21	0.00
20	41	12.49	0.00	70.93	0.00	161.41	0.00	8.51	0.00
21	44	6.04	∞	16.42	4805.93	151.84	1019.41	15.57	0.56
22	47	18.03	∞	107.73	∞	16.69	∞	23.72	24.31
23	50	47.63	∞	154.47	∞	367.83	∞	6.94	2269.73
24	54	55.45	0.00	54.28	0.00	80.74	0.00	12.47	0.00
25	57	8.04	∞	96.56	∞	655.66	∞	59.70	∞
26	61	13.79	0.00	62.72	0.00	168.90	0.00	24.41	0.00
27	65	42.04	0.00	5590.12	0.00	193.44	0.00	23.62	0.00
28	68	∞	∞	∞	∞	∞	∞	54.81	∞
29	72	∞	0.00	∞	0.00	∞	0.00	∞	0.00
30	76	∞	0.00	∞	0.00	∞	0.00	∞	0.00
31	80	∞	∞	∞	∞	∞	∞	21.95	1999.14
32	85	∞	0.00	5170.12	0.00	∞	0.00	80.52	0.00

the timeout period – for example geng takes over 60 hours to find $F_4(20)$. We stress though that the goal of our experiments is to demonstrate the benefit of our symmetry breaks, not to compare the speed of our approach to other algorithms.

We present the results obtained using BEE which compiles finite domain constraints to CNF and solves them using an underlying SAT solver. Our configuration uses CryptoMiniSat v2.5.1 [32]. BEE performs CNF simplification by applying a constraint-driven technique called equi-propagation [33] and partial evaluation. All experiments are performed on a single core of an Intel(R) Core(TM) i5-2400 3.10GHz CPU with 4GB memory under Linux (Ubuntu lucid, kernel 2.6.32-24-generic). BEE is written in Prolog and run using SWI Prolog v6.0.2 64-bits. All experiments were replicated and verified using the Choco constraint programming toolkit. As run times were considerably higher in this case we only give results for the BEE experiments.

4.1 Experiment 1: computing $f_4(v)$

Table 1 summarizes the results for a constraint-based approach to compute values of $f_4(v)$. We compare the computation time for four configurations: (1) “no symmetry break” break, (2) breaking symmetries using sb_ℓ, (3) breaking symmetries using sb^*_ℓ, and (4) breaking symmetries using sb_ℓ with an embedded star. The columns in Table 1 specify (from left to right): The number of vertices, v, and the value $f_4(v)$. Then, for each of the four configurations, we specify the computation time to compute a graph with $\mathrm{f}_4(v)$ edges (columns labeled “sat”), and show the non-existence of a graph with $\mathrm{f}_4(v)+1$ edges (columns labeled “unsat”).

For the first three configurations, we apply the constraint model from Fig. 11. For the fourth configuration, we add an additional constraint to the model. To this end we follow [9] where it is noted that every graph in $\mathcal{F}(v)$ with at least 5 vertices and minimum/maximum vertex degrees (δ, Δ) contains a $(\Delta, \delta-1)$-star. In general, an (m, n)-star is a rooted tree, denoted $S_{m,n}$, where the root has m children, each of which has $n \geq 1$ children, all of which are leaves. The existence of a $S_{\Delta,\delta-1}$ in *any* extremal graph (with at least 5 vertices) follows from the fact that the children and grandchildren of any vertex are distinct (as there are no 3− or 4−cycles). So, we add constraints to explicitly embed $S_{\Delta,\delta-1}$ in the adjacency matrix. In Section 4.2, in order to distinguish $S_{\Delta,\delta-1}$ stars from new stars that we introduce, we refer to a *simple Garnick star*. In this setting, based on Theorem 6 we impose symmetry breaking on the Δ clusters of $\delta-1$ leaves of $S_{\Delta,\delta-1}$ as well as to the cluster of vertices not

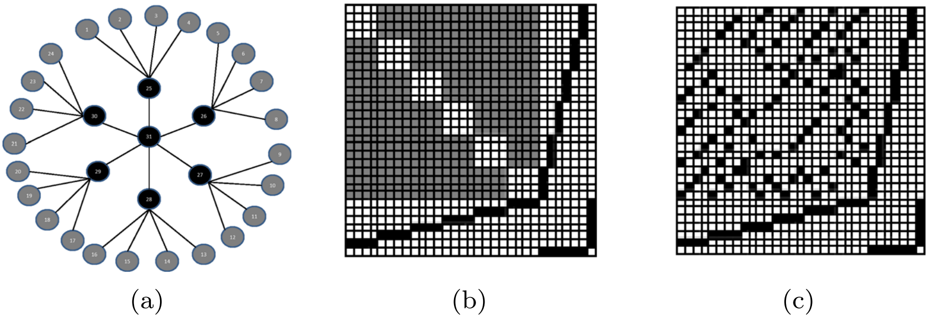


Fig. 12 **a** the star $S_{6,4}$, **b** its adjacency matrix, and **c** a member of $\mathcal{F}_4(31)$ with 80 edges (0=white, 1=black, not determined = gray)

in the star (and use the partition preserving symmetry break as described in Theorem 6). Figure 12a illustrates the star $S_{6,4}$.

A 31 × 31 adjacency matrix with an embedded $S_{6,4}$ is depicted as Fig. 12b. Black and white cells indicate values 1 and 0 respectively, and gray cells indicate unassigned Boolean variables. The last row of the matrix corresponds to the root. Moving up we find the 6 children of the root, and then its 24 grandchildren. Note that although in this particular example all vertices are in the star, this is not generally true. We will later return to explain Fig. 12c.

Examining the first three columns labeled "sat" it appears that there is no significant gain in symmetry breaking when the instance is satisfiable and we need only find a single witness. In fact for many of the instances, the computation with no symmetry break is faster. When instances are unsatisfiable (the "unsat" columns), we encounter two types of instances: those which involve search and those which do not. For the later type, unsatisfiability derives from the propagation of the constraints in Eq. 1 and the computation is fast for all configurations. For the other instances, the solver must explore the entire search space and the first three "unsat" columns indicate that symmetry breaking is then useful.

The bottom two rows in Table 1 describe our results for two open instances, computing $f_4(31)$ and $f_4(32)$. A lower bound of $f_4(31) \geq 80$ is given in [9] and a witness (discovered using our model in less than 22 seconds) is depicted as Fig. 12c. It is canonical with respect to a partitioning where the first 24 rows form 6 clusters of size 4 each (the grandchildren), the next 6 rows form a cluster (the children), and the last row is a singleton cluster (the root). With the proof that there is no witness with 81 edges (determined using our model in 33 minutes of CPU time) we conclude that $f_4(31) = 80$. Given that $f_4(31) = 80$, Eq. 1 implies that $f_4(32) \leq 85$ and hence that the lower bound $f_4(32) \geq 85$ reported in [9] is the precise value, consequently $f_4(32) = 85$. These are both new results.

A comparison between the configuration sb_ℓ and sb_ℓ with an embedded star demonstrates that the search times when considering the embedded stars are much faster.

4.2 Experiment 2: computing $F_4(v)$

In this experiment we apply a constraint-based approach to compute the number of non-isomorphic extremal graphs with v vertices. We apply a constraint solver to generate all graphs satisfying the constraint model for v vertices and $e = f_4(v)$ edges with corresponding symmetry breaking constraints. We then apply `nauty` to determine the number of non-isomorphic graphs within this set. The time required to run `nauty` is negligible and not detailed in our results.

For smaller values, $v \leq 15$, we consider the constraint model of Fig. 11. Table 2 shows for each value of v the maximum number of edges $f_4(v)$, the number of non-isomorphs $F_4(v)$, and the number of graphs generated (columns $sols$) and computation time ($time$, in seconds), for each of the three configurations. Our results are as expected: improving symmetry breaking makes a significant difference.

For $16 \leq v \leq 19$, for all configurations we again follow [9] and apply an additional constraint to embed a $(\Delta, \delta - 1)$ star (as we did for the final configuration in Table 1). An interesting observation is that for $17 \leq v \leq 19$ the number of graphs generated is the same for sb_ℓ and sb^*_ℓ. This is due to the structure of the solutions in these cases. Note that for sb^*_ℓ to generate fewer graphs than sb_ℓ, for some partition P_k and $\{i, j\} \subseteq P_k$ with $i < j$, $A[i] \preceq A[j]$ but $A[i] \npreceq_{\{i,j\}} A[j]$. This can only happen if (i, j) is an edge, and so can only occur in the partition containing the vertices that are not in the star. For the case $v = 17$ and $(\delta, \Delta) = (4, 5)$ there is only one vertex not contained in the star and so this is not possible. The other cases require further analysis but can be explained in a similar way.

Table 2 Computing $F_4(v)$ (time in seconds; timeout 4hrs)

v	f$_4$(v)	F$_4$(v)	no sym break		sb_ℓ		sb^*_ℓ	
			# sols	time	# sols	time	# sols	time
4	3	2	9	0.01	3	0.03	2	0.01
5	5	1	12	0.01	1	0.01	1	0.01
6	6	2	120	0.05	4	0.03	2	0.03
7	8	1	900	0.25	6	0.01	3	0.01
8	10	1	2520	1.95	4	0.02	1	0.02
9	12	1	10080	16.48	6	0.06	3	0.06
10	15	1	30240	48.96	2	0.03	1	0.04
11	16	3	—	∞	48	0.91	16	0.74
12	18	7	—	∞	469	1.93	192	0.75
13	21	1	—	∞	66	0.21	27	0.15
14	23	4	—	∞	2888	81.67	1021	16.02
15	26	1	—	∞	812	51.72	268	4.70
16	28	22	—	∞	—	∞	—	∞
17	31	14	—	∞	29289	1249.66	29289	981.92
18	34	15	—	∞	47811	11321.10	47811	7469.56
19	38	1	49248	430.58	70	0.14	70	0.14
20	41	1	—	∞			14	2.57
21	44	3	—	∞			438	1.90
22	47	3	—	∞			372	302.42
23	50	7	—	∞			2220	791.42
24	54	1	—	∞			144	28.48
25	57	6	—	∞			7740	10791.31
32	85	1	—	∞			240	765.16

For the larger instances where $v \geq 20$, we first extend the approach regarding embedding a star in extremal graphs. We denote by $S_{m,[n_1,\ldots,n_m]}$, a rooted tree with m children, which respectively have $n_1, \ldots, n_m$ children (grandchildren of the root). In [13], for each $v \geq 21$ and feasible pair (δ, Δ) we have identified stars which *must* be contained in any extremal graph (see Proposition 1). We embed the indicated structure in the adjacency matrix when performing the encoding using clusters in a similar way to that used previously using simple Garnick embedded stars (and again use the partition preserving symmetry break as described in Theorem 6). Note that we have identified stars containing as many vertices as possible, so that our "not in the star" cluster is a small as possible.

The bottom rows in Table 2 describe our results for four open instances, computing $F_4(22)$, $F_4(23)$, $F_4(24)$, $F_4(25)$ and $F_4(32)$. These are new results and we did not rely on any previous known bounds to reduce the search space. Because of the embedded star, there is no noticeable difference between the symmetry break using sb^*_ℓ or when using sb_ℓ (and so we only present the results for sb^*_ℓ in the table). However, as indicated by the table, not applying either is catastrophic. The single element of $F_4(32)$ is shown in Fig. 13.

Note that values of $F_4(v)$ for $26 \leq v \leq 31$ have also been calculated (and presented in [13]). In this case, although our approach was used to eliminate some sub-cases, the

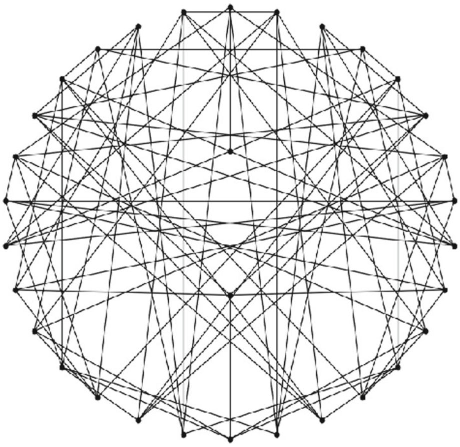


Fig. 13 Unique girth 5 graph of order 32 with 85 edges

graphs were largely constructed by hand from graphs in $F_4(v-1)$, so we do not include experimental results here. Note that the values of $F_4(v)$ in these cases are however included in Table 3.

Table 3 Embedded stars for $20 \le v \le 25$, and $v = 32$

v	$f_4(v)$	(δ, Δ)	Form of the embedded star
20	41	(3, 5)	$S_{5,[3,3,3,3,2]}$
21	44	(3, 5)	$S_{5,[3,3,3,3,3]}$
		(4, 5)	$S_{5,[3,3,3,3,3]}$
22	47	(3, 5)	$S_{5,[4,3,3,3,3]}$
		(4, 5)	$S_{5,[4,3,3,3,3]}$
23	50	(3, 5)	$S_{5,[4,4,3,3,3]}$
		(4, 5)	$S_{5,[4,4,3,3,3]}$ or $S_{5,[4,3,3,3,3]}$
24	54	(4, 5)	$S_{5,[4,4,4,3,3]}$
25	57	(3, 5)	$S_{5,[4,4,4,4,3]}$ or $S_{5,[4,4,4,3,3]}$
		(4, 5)	$S_{5,[4,4,4,4,3]}$ or $S_{5,[4,4,4,3,3]}$
		(4, 6)	$S_{6,[3,3,3,3,3,3]}$
26	61	(4, 5)	$S_{5,[4,4,4,4,4]}$
27	65	(4, 5)	$S_{5,[4,4,4,4,4]}$
28	68	(3, 6)	$S_{6,[4,4,4,4,3,2]}$
		(4, 5)	$S_{5,[4,4,4,4,4]}$
		(4, 6)	$S_{6,[4,4,4,3,3,3]}$
29	72	(4, 6)	$S_{6,[4,4,4,4,3,3]}$
30	76	(4, 6)	$S_{6,[4,4,4,4,4,3]}$ and $S_{6,[5,4,4,4,3,3]}$
31	80	(4, 6)	$S_{6,[4,4,4,4,4,4]}$ and $S_{6,[5,4,4,4,4,3]}$
		(5, 6)	$S_{6,[4,4,4,4,4,4]}$
32	85	(5, 6)	$S_{6,[5,4,4,4,4,4]}$

Proposition 1 *Let G be an extremal graph with v vertices where $20 \leq v \leq 27$ or $v = 32$. Then, the minimal and maximal degrees, (δ, Δ) of a vertex in G correspond to one of the cases indicated in* Table 3. *Furthermore, G has an embedded star of one of the forms indicated in* Table 3 *for the respective values of (δ, Δ). Note that if more than one star is indicated then either all graphs have embedded stars of at least one of the indicated types (where the form is given as $S_{m,[n_1,\dots,n_m]}$ or $S_{m',[n'_1,\dots,n'_m]}$), or all graphs have both of the indicated types (where the form is given as $S_{m,[n_1,\dots,n_m]}$ and $S_{m',[n'_1,\dots,n'_m]}$).*

The full proof of Proposition 1 is presented in [13].

4.3 Experiment 3: computing $f_5(v)$

For our final experiment regarding girth we consider the extremal graphs which contain no cycles of length 5 or less. To this end we extend the basic constraint model of Fig. 11 with an additional constraint that states that every sequence of five vertices does not form a cycle, and we consider the optimization problem which computes values of $f_5(v)$. Table 4 shows our results. To better illustrate the impact of improved lexicographic symmetry breaking we consider also a predicate sb^+_ℓ which is like sb^*_ℓ but only compares consecutive rows of the matrix. The reason we did this was to satisfy our curiosity as to whether the additional work required for sb^*_ℓ ($O(n^2)$ compared to $O(n)$) is worth it. It is clear that the much larger sb^*_ℓ pays off. The stricter conditions force partial solutions to be rejected quicker, reducing computation time considerably for the larger instances.

5 Symmetry breaking for graph colorings

In this section we show that all of our results carry over to the search for graph edge colorings where the adjacency matrix contains integer values representing the color of edges instead of Boolean values representing the presence of edges. A graph coloring, in k colors,

Table 4 Computing $f_5(v)$ (time in seconds; timeout 4hrs)

v	f$_5$(v)	No sym break	sb_ℓ	sb^+_ℓ	sb^*_ℓ
4	3	0.00	0.00	0.01	0.02
5	4	0.01	0.01	0.01	0.01
6	6	0.01	0.01	0.01	0.01
7	7	0.14	0.03	0.03	0.02
8	9	2.01	0.06	0.04	0.04
9	10	76.07	0.13	0.08	0.07
10	12	2224.65	0.30	0.02	0.15
11	14	∞	1.29	0.61	0.41
12	16	∞	4.97	3.49	1.30
13	18	∞	21.18	11.19	7.93
14	21	∞	85.73	43.78	20.29
15	22	∞	801.03	418.05	203.08
16	24	∞	∞	6076.86	1613.60
17	26	∞	∞	∞	13903.10

$$\forall_{1\leq i<j\leq n}.\ \big(1 \leq A[i,j] \leq 3 \wedge A[i,j] = A[j,i]\big) \quad \wedge \quad \forall_{1\leq i\leq n}.\ \big(A[i,i] = 0\big)$$

$$\forall_{1\leq i<j<k\leq n}.\ \neg\big(A_{i,j} = A_{j,k} = A_{k,i}\big)$$

Fig. 14 Constraint model for Ramsey colorings (3, 3, 3; n)

is a pair (G, κ) consisting of a simple graph $G = (V, E)$ and a mapping $\kappa\colon E \to \{1, \dots, k\}$. We typically represent (G, κ) with $|V| = \{1, \dots, n\}$, as an $n \times n$ adjacency matrix, A, defined such that

$$A[i, j] = \begin{cases} \kappa(i, j) & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

A graph coloring problem is a formula $\varphi(A)$ where A is an $n \times n$ adjacency matrix of integer variables (represented edge colors) together with a set (conjunction) of constraints φ on these variables. A solution is an assignment of integer values to the variables in A which satisfy φ and determine both the graph edges and their colors.

Suppose we are looking for a graph coloring A that satisfies a given property P. Here, solutions are typically closed under permutations both of vertices and of colors. Restricting the search space for a solution modulo such permutations is crucial when trying to solve hard graph coloring problems.

We observe that the notion of a canonical form given as Definition 5 and the lexicographic symmetry breaking predicates sb_ℓ and sb^*_ℓ of Definitions 6 and 8 are well-defined also for graph colorings where the non-zero entries in an adjacency matrix have values other than 1. Moreover, the proofs of Theorems 1 and 4 do not rely on the non-zero entries in an adjacency matrix having only value 1. Therefore our symmetry breaking constraints can be applied also in the search for graph colorings. We illustrate the impact of our approach on the search for Ramsey colorings (3, 3, 3; n) of the complete graph K_n.

An $(r_1, \dots, r_k; n)$ Ramsey coloring is an assignment of one of k colors to each edge in the complete graph K_n such that it does not contain a monochromatic complete subgraph K_{r_i} in color i for $1 \leq i \leq k$. The Ramsey number $R(r_1, \dots, r_k)$ is the least $n > 0$ such that no $(r_1, \dots, r_k; n)$ coloring exists. The only known value of a nontrivial multicolor classical Ramsey number is $R(3, 3, 3) = 17$. The number of (3, 3, 3; n) colorings of the complete graph K_n on n vertices is known for $14 \leq n \leq 16$ [14]. We illustrate here that we can reproduce these numbers using the symmetry break techniques proposed in this paper. Unfortunately, our technique does not suffice to compute the number of (3, 3, 3; 13) colorings.

Figure 14 illustrates the constraints of the Ramsey coloring problem (3, 3, 3; n). Constraint (8) states that the graph has n vertices, is 3 colored, and is simple (symmetric, and with no self loops). Constraint (9) states that the n vertex graph has no embedded monochromatic sub-graph K_3.

Table 5 illustrates the impact of the symmetry breaking predicate sb^* on the search for (3, 3, 3; n) Ramsey colorings. The column headed by "$\#\backslash_{\approx}$" specifies the known number of colorings modulo weak isomorphism [14].

Definition 13 ((weak) isomorphism of graph colorings) Let (G, κ_1) and (H, κ_2) be k-color graph colorings with $G = ([n], E_1)$ and $H = ([n], E_2)$. We say that (G, κ_1) and (H, κ_2) are weakly isomorphic, denoted $(G, \kappa_1) \approx (H, \kappa_2)$ if there exist permutations $\pi\colon [n] \to [n]$ and $\sigma\colon [k] \to [k]$ such that $(u, v) \in E_1 \iff (\pi(u), \pi(v)) \in E_2$ and

Table 5 The search for $(3, 3, 3; n)$ Ramsey colorings with and without the symmetry break sb* (time in seconds with 24 hr. timeout)

n	$\#\backslash_{\approx}$	No sym break			With sym break			
		#vars	#clauses	time	#vars	#clauses	time	#
17	0	408	2584	3042.10	4038	20734	**0.15**	0
16	2	360	2160	∞	3328	17000	**0.14**	6
15	2	315	1785	∞	2707	13745	**0.37**	66
14	115	273	1456	∞	2169	10936	**259.560**	24635
13	—	234	1170	∞	1708	8540	∞	—

$\kappa_1((u, v)) = \sigma(\kappa_2((\pi(u), \pi(v))))$. When σ is the identity permutation, we say that (G, κ_1) and (H, κ_2) are isomorphic.

The columns headed by "#vars" and "#clauses" indicate, respectively, the number of variables and clauses in the corresponding CNF encodings of the coloring problems with and without the symmetry breaking constraint. The columns headed by "time" indicate the time (in seconds) to find all colorings iterating with a SAT solver. The timeout assumed here is 24 hours. The column headed by "#" specifies the number of colorings found when solving with the symmetry break. These include colorings which are weakly isomorphic, but far less than the hundreds of thousands generated without the symmetry break (until the timeout). We have verified using `nauty` that the colorings obtained using the symmetry break (the last column) reduce modulo isomorphism to the known numbers (the second column).

Figure 15 presents the two non-isomorphic colorings $(3, 3, 3; 16)$ represented as adjacency graphs in the form found using our encoding. Note the lexicographic order on the rows in both matrices. These graphs are isomorphic to the two colorings reported in 1968 by Kalbfleish and Stanton [34] where it is also proven that there are no others (modulo weak isomorphism).

There are many papers that consider ways to exploit symmetries in the search for Ramsey numbers and colorings. In particular: Gent and Smith [35], building on the work of Puget [36], study symmetries in graph coloring problems and recognize the importance of breaking symmetries during search. Meseguer and Torras [37] present a framework for exploiting symmetries to heuristically guide a depth first search, and also show results for $(3, 3, 3; n)$ Ramsey colorings with $14 \leq n \leq 17$. An advantage of our approach is that it is

$$\begin{bmatrix}
\mathbf{0}&1&1&1&1&1&2&2&2&2&2&3&3&3&3&3\\
1&\mathbf{0}&2&2&3&3&1&1&2&2&3&1&1&2&3&3\\
1&2&\mathbf{0}&3&2&3&2&3&1&1&2&1&2&3&1&3\\
1&2&3&\mathbf{0}&3&2&1&2&1&3&2&2&3&1&3&1\\
1&3&2&3&\mathbf{0}&2&3&2&2&1&1&3&1&3&2&1\\
1&3&3&2&2&\mathbf{0}&2&1&3&2&1&3&3&1&1&2\\
2&1&2&1&3&2&\mathbf{0}&3&3&1&1&2&3&2&1&3\\
2&1&3&2&2&1&3&\mathbf{0}&1&1&3&3&2&2&3&1\\
2&2&1&1&2&3&3&1&\mathbf{0}&3&1&2&1&3&3&2\\
2&2&1&3&1&2&1&1&3&\mathbf{0}&3&3&2&1&2&3\\
2&3&2&2&1&1&1&3&1&3&\mathbf{0}&1&3&3&2&2\\
3&1&1&2&3&3&2&3&2&3&1&\mathbf{0}&2&1&2&1\\
3&1&2&3&1&3&3&2&1&2&3&2&\mathbf{0}&1&1&2\\
3&2&3&1&3&1&2&2&3&1&3&1&1&\mathbf{0}&2&2\\
3&3&1&3&2&1&1&3&3&2&2&2&1&2&\mathbf{0}&1\\
3&3&3&1&1&2&3&1&2&3&2&1&2&2&1&\mathbf{0}
\end{bmatrix}
\qquad
\begin{bmatrix}
\mathbf{0}&1&1&1&1&1&2&2&2&2&2&3&3&3&3&3\\
1&\mathbf{0}&2&2&3&3&1&1&2&2&3&1&1&2&3&3\\
1&2&\mathbf{0}&3&2&3&1&2&1&3&2&2&3&1&1&3\\
1&2&3&\mathbf{0}&3&2&2&2&3&1&1&3&1&1&3&2\\
1&3&2&3&\mathbf{0}&2&2&3&1&2&1&1&3&3&2&1\\
1&3&3&2&2&\mathbf{0}&3&1&2&1&2&3&2&3&1&1\\
2&1&1&2&2&3&\mathbf{0}&3&3&1&1&2&3&2&3&1\\
2&1&2&2&3&1&3&\mathbf{0}&1&3&1&3&2&1&2&3\\
2&2&1&3&1&2&3&1&\mathbf{0}&1&3&2&1&3&2&3\\
2&2&3&1&2&1&1&3&1&\mathbf{0}&3&1&2&3&3&2\\
2&3&2&1&1&2&1&1&3&3&\mathbf{0}&3&3&2&1&2\\
3&1&2&3&1&3&2&3&2&1&3&\mathbf{0}&2&1&1&2\\
3&1&3&1&3&2&3&2&1&2&3&2&\mathbf{0}&2&1&1\\
3&2&1&1&3&3&2&1&3&3&2&1&2&\mathbf{0}&2&1\\
3&3&1&3&2&1&3&2&2&3&1&1&1&2&\mathbf{0}&2\\
3&3&3&2&1&1&1&3&3&2&2&2&1&1&2&\mathbf{0}
\end{bmatrix}$$

Fig. 15 $R(3, 3, 3; 16)$ graphs

not specialized. We simply require the rows of the adjacency matrix to be lexicographically ordered.

In [15] we apply the results of this paper to show that the precise value of $R(4, 3, 3)$ is 30. This was an open problem for over 30 years.

6 Conclusion

We have considered the problem of breaking symmetries during search when identifying undirected graphs satisfying a given property P, assuming a setting where testing for the existence of a graph G satisfying P is posed as a Boolean constraint $P(A_G)$ on the variables of the Boolean adjacency matrix A_G of G. We have presented two symmetry breaking constraints (sb_ℓ and sb^*_ℓ) and formally proved their correctness. We have demonstrated the benefit of our approach by applying it to solve a variety of problems related to extremal graphs of girth at least $k+1$, for $k = 4$ and $k = 5$, solving some open instances. In particular we have shown how combining our technique with known properties of extremal graphs (in our case, the existence of embedded stars) can increase its effectiveness. We have also extended the approach to apply to graph coloring problems and demonstrated its impact on the search for Ramsey numbers and coloring.

Funding Information Codish was supported by the Israel Science Foundation grant 625/17.

References

1. Washio, T., & Motoda, H. (2003). State of the art of graph-based data mining. *SIGKDD Explorations*, *5*(1), 59–68.
2. Faulon, J.-L. (1998). Isomorphism, automorphism partitioning, and canonical labeling can be solved in polynomial-time for molecular graphs. *Journal of Chemical Information and Computer Sciences*, *38*(3), 432–444.
3. Ehrlich, H., & Rarey, M. (2011). Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *1*(1), 68–79.
4. Solnon, C., Damiand, G., de la higuera, C., Janodet, J.. (2015). On the complexity of submap isomorphism and maximum common submap problems. *Pattern Recognition*, *48*(2), 302–316.
5. Gardiner, E.J., & Sakr, S. (2011). Graph applications in chemoinformatics and structural bioinformatics. In Pardede, E. (Ed.) *Graph Data Management* (pp. 386-420). Pennsylvania: IGI Global.
6. McKay, B. (1990). Nauty user's guide (version 1.5), Tech. Rep. TR-CS-90-02, Australian National University Computer Science Department.
7. McKay, B., & Piperno, A. (2014). Practical graph isomorphism, II. *Journal of Symbolic Computation*, *60*, 94–112.
8. Bollobás, B. (1978). Extremal graph theory, Courier Dover Publications.
9. Garnick, D., Kwong, Y., Harris, F. (1993). Lazebnik, Extremal graphs without three-cycles or four-cycles. *Journal of Graph Theory*, *17*(5), 633–645.

10. Crawford, J.M., Ginsberg, M.L., Luks, E.M., Roy, A. (1996). Symmetry-breaking predicates for search problems. In: Aiello, C.L., Doyle, J., Shapiro, S. C. (Eds.), KR, Morgan Kaufmann, pp. 148–159.
11. Shlyakhter, I. (2007). Generating effective symmetry-breaking predicates for search problems. *Discrete Applied Mathematics*, *155*(12), 1539–1548.
12. Codish, M., Miller, A., Prosser, P., Stuckey, P. (2013). Breaking symmetries in graph representation. In: Proceedings of the international joint conference on artificial intelligence (IJCAI 2013), Beijing, China, pp. 510–516.
13. Miller, A., & Codish, M. (2017). Graphs of girth 5 with orders between 20 and 32. arXiv:1708.06576 [math.CO].
14. Radziszowski, S. Small Ramsey numbers, dynamic survey, revision #14, The Electronic Journal of Combinatorics.
15. Codish, M., Frank, M., Itzhakov, A., Miller, A. (2016). Computing the Ramsey number R(4,3,3) using abstraction and symmetry breaking. *Constraints*, *21*(3), 375–393.
16. Read, R.C. (1978). Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations. *Annals of Discrete Mathematics*, *2*, 107–120.
17. Miller, A., & Prosser, P. (2012). Diamond-free degree sequences. *Acta Univ. Sapientiae, Informatica*, *4*(2), 189–200.
18. Gent, I., Prosser, P., Smith, B. (2002). A 0/1 encoding of the gaclex constraint for pairs of vectors. In: ECAI 2002 Workshop w9: Modelling and solving problems with constraints.
19. Flener, P., Frisch, A.M., Hnich, B., Kiziltan, Z., Miguel, I., Pearson, J., Walsh, T. (2002). Breaking row and column symmetries in matrix models. In: Hentenryck, P.V. (Eds.), CP, vol. 2470 of Lecture Notes in Computer Science, Springer, pp. 462–476.
20. Frisch, A.M., Jefferson, C., Miguel, I. (2003). Constraints for breaking more row and column symmetries. In: Rossi, F. (Eds.), CP, Vol. 2833 of Lecture Notes in Computer Science, Springer, pp. 318–332.
21. Ou, J. (2007). On extremal unicyclic molecular graphs with prescribed girth and minimal hosoya index. *Journal of Mathematical Chemistry*, *42*(3), 423–432.
22. Kincaid, R.K., Kunkler, S.J., Lamar, M.D., Phillips, D.J. Algorithms and complexity results for finding graphs with extremal randić index, Networks.
23. Polak, M., Romańczuk, U., Ustimenko, V., Wróblewska, A. (2013). On the applications of extremal graph theory to coding theory and cryptography. *Electronic Notes in Discrete Mathematics*, *43*, 329–342.
24. Abajo, E., & Diánez, A. (2010). Exact values of ex(v;{C_3, C_4, ..., C_n}). *Discrete Applied Mathematics*, *158*(17), 1869–1878.
25. Garnick, D., & Nieuwejaar, A. (1992). Non-isomorphic extremal graphs without three-cycles or four-cycles. *Journal of Combinatorial Mathematics and Combinatorial Computing*, *12*, 33–56.
26. Backelin, J. (2015). Sizes of extremal girth 5 graphs of orders from 40 to 49. arXiv:1511.08128[math.CO].
27. Wang, P., Dueck, G.W., MacMillan, S. (2001). Using simulated annealing to construct extremal graphs. *Discrete Mathematics*, *235*(1-3), 125–135.
28. Abajo, E., Balbuena, C., Diánez, A. (2010). New families of graphs without short cycles and large size. *Discrete Applied Mathematics*, *158*(11), 1127–1135.
29. Bong, N.H. Some new upper bounds of $ex(n; \{C_3, C_4\})$, AKCE International Journal of Graphs and Combinatorics. Online edition at https://doi.org/10.1016/j.akcej.2017.03.006.
30. (2010). The on-line encyclopedia of integer sequences, published electronically at http://oeis.org.
31. Metodi, A., & Codish, M. (2012). Compiling finite domain constraints to SAT with BEE. *TPLP*, *12*(4-5), 465–483.
32. Soos, M. (2010). CryptominiSAT, v2.5.1 http://www.msoos.org/cryptominisat2.
33. Metodi, A., Codish, M., Lagoon, V., Stuckey, P.J.. (2011). Boolean equi-propagation for optimized SAT encoding. In: Lee, J. H.-M. (Eds.), CP, Vol. 6876 of Lecture Notes in Computer Science, Springer, pp. 621–636.
34. Kalbfleisch, J., & Stanton, R. (1968). On the maximal triangle-free edge-chromatic graphs in three colors. *Journal of Combinatorial Theory*, *5*, 9–20.
35. Gent, I.P., & Smith, B.M. (2000). Symmetry breaking in constraint programming. In Horn, W. (Ed.) *ECAI 2000, Proceedings of the 14th European conference on artificial intelligence* (pp. 599–603). Berlin: IOS Press.
36. Puget, J. (1993). On the satisfiability of symmetrical constrained satisfaction problems. In Komorowski, H.J., & Ras, Z.W. (Eds.) *Methodologies for intelligent systems, 7th international symposium, ISMIS '93* (pp. 350–361). Trondheim: Springer. Proceedings, vol. 689 of lecture notes in computer science.
37. Meseguer, P., & Torras, C. (2001). Exploiting symmetries within constraint satisfaction search. *Artif Intell.*, *129*(1-2), 133–163.